

**TUGAS AKHIR - KS 141501**

**OPTIMASI PENJADWALAN UJIAN OTOMATIS  
DENGAN MENGGUNAKAN ALGORITMA GREEDY  
- SIMULATED ANNEALING - HYPER HEURISTIC**

**DIAN KUSUMAWARDANI**  
NRP 5210 100 127

Dosen Pembimbing  
Ahmad Muklason, S.Kom., M.Sc., Ph.D.

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018

FINAL PROJECT - KS 141501

***AUTOMATED EXAMINATION TIMETABLING  
OPTIMIZATION USING GREEDY - SIMULATED  
ANNEALING - HYPER HEURISTIC ALGORITHM***

DIAN KUSUMAWARDANI  
NRP 5210 100 127

Supervisors  
Ahmad Muklason, S.Kom., M.Sc., Ph.D.

DEPARTMENT OF INFORMATION SYSTEMS  
Faculty of Information Technology and Communication  
Sepuluh Nopember Institute of Technology  
Surabaya 2018

**TUGAS AKHIR - KS141501**

# **OPTIMALISASI PENJADWALAN UJIAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA GREEDY - SIMULATED ANNEALING - HYPER-HEURISTIC**

**DIAN KUSUMAWARDANI**  
**NRP 5210 100 127**

**Dosen Pembimbing**  
**Ahmad Muklason, S.Kom., M.Sc., Ph.D.**

**JURUSAN SISTEM INFORMASI**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2018**



**FINAL PROJECT - KS141501**

# **AUTOMATED EXAMINATION TIMETABLING OPTIMIZATION USING GREEDY GREEDY - SIMULATED ANNEALING - HYPER-HEURISTIC**

**DIAN KUSUMAWARDANI**  
**NRP 5210 100 127**

**Supervisors**  
**Ahmad Muklason, S.Kom., M.Sc., Ph.D.**

**INFORMATION SYSTEM DEPARTMENT**  
**Faculty of Information Technology**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2018**



## LEMBAR PENGESAHAN

### OPTIMASI PENJADWALAN UJIAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA GREEDY – SIMULATED ANNEALING – HYPER HEURISTIC

#### TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**DIAN KUSUMAWARDANI**  
**NRP 5210 100 127**



Surabaya, 17 Januari 2018

**PLH Kepala**  
**Departemen Sistem Informasi**

**Edwin Riksakomara, S.Kom, M.T.**  
**NIP 19690725 200312 1 001**





## **LEMBAR PERSETUJUAN**

### **OPTIMASI PENJADWALAN UJIAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA GREEDY – LATE ACCEPTANCE – HYPER HEURISTIC**

#### **TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**DIAN KUSUMAWARDANI**  
**5210 100 127**

Disetujui Tim Penguji: Tanggal Ujian : Januari 2018  
Periode Wisuda : Maret 2018

**Ahmad Muklason, S.Kom., M.Sc., Ph.D. (Pembimbing I)**

**Wiwik Anggraeni, S.Si., M.Kom.**

**(Penguji I)**

**Edwin Riksakomara, S.Kom., M.T.**

**(Penguji II)**



# **OPTIMASI PENJADWALAN UJIAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA GREEDY – SIMULATED ANNEALING – HYPER HEURISTIC**

**Nama Mahasiswa** : DIAN KUSUMAWARDANI  
**NRP** : 5210 100 127  
**Jurusan** : SISTEM INFORMASI FTIF-ITS  
**Dosen Pembimbing** : Ahmad Muklason, S.Kom., M.Sc.,  
Ph.D.

## **ABSTRAK**

*Bagi sebuah lembaga pendidikan penjadwalan adalah salah satu kegiatan utama yang dilakukan pada setiap tahun ajaran. Salah satu jadwal yang dibuat adalah jadwal ujian, dimana ujian adalah salah satu hal yang penting karena merupakan salah satu cara yang dilakukan untuk mengukur kompetensi peserta didik sehingga kegiatan akademik ini wajib untuk diikuti oleh setiap peserta didik, baik itu pada tengah semester maupun akhir semester. Jadwal ujian yang telah disusun untuk masing-masing mahasiswa haruslah terbebas dari jadwal yang bentrok sehingga setiap mahasiswa dapat mengikuti seluruh ujian untuk mata kuliah yang mereka ikuti*

*Pada tugas akhir dihasilkan sebuah sistem penjadwalan ujian yang mampu menghasilkan jadwal ujian secara otomatis dengan menggunakan algoritma greedy – simulated annealing – hyper heuristic yang dibangun dengan menggunakan bahasa pemrograman java.*

*Hasil yang didapat menunjukkan bahwa algoritma greedy – simulated annealing – hyperheuristic dapat digunakan untuk melakukan penjadwalan ujian baik itu UTS maupun UAS dan juga jadwal yang dihasilkan lebih optimal apabila dibandingkan dengan jadwal yang dihasilkan secara manual dimana dibuktikan dengan berhasil menurunkan nilai proximity cost sebesar 39,569 yang didapat dari jadwal manual menjadi 33,649. Diharapkan dengan adanya pengerjaan tugas*

*akhir ini dapat membantu pihak Departemen Sistem Informasi dalam penyusunan jadwal ujian yang lebih baik.*

***Kata kunci: greedy, hyper-heuristic, java, optimal, penjadwalan, simulated annealing, ujian***

# **AUTOMATED EXAMINATION TIMETABLING OPTIMIZATION USING GREEDY – SIMULATED ANNEALING – HYPER HEURISTIC ALGORITHM**

**Name** : DIAN KUSUMAWARDANI  
**NRP** : 5210 100 127  
**Departement** : INFORMATION SYSTEM FTIF-ITS  
**Supervisors** : Ahmad Muklason, S.Kom., M.Sc., Ph.D.

## **ABSTRACT**

*For an educational institution, scheduling is one of the main activities undertaken in each academic year. One of the schedules made is the exam schedule, where the exam is one of the important things because it is one way that is done to measure the competence of learners so that this academic activity must be followed by every learner, either in the middle semester or the end of the semester. The exam schedules that have been prepared for each student must be free from conflicting schedules so that every student can take all the exams for the courses they follow*

*In the final task generated an exam scheduling system that is able to generate test schedules automatically by using greedy - simulated annealing - hyper heuristic algorithm that is built using java programming language.*

*The results show that the greedy-simulated annealing-hyperheuristic algorithm can be used to schedule the test both UTS and UAS as well as the resulting schedule is more optimal when compared with the manually generated schedule which is successfully reduced the value of proximity cost of 39.569 obtained from manual to 33,649. It is expected that this final project will help the Department of Information System in preparing a better test schedule.*

-

***Kata kunci: exam, greedy, hyper-heuristic, java, optimal, scheduling, simulated annealing***

## **KATA PENGANTAR**

Puji syukur penulis ucapkan ke hadirat Allah SWT yang telah memberikan segala petunjuk, kemudahan, dan kelancaran bagi penulis untuk menyelesaikan pengerjaan tugas akhir dengan judul:

### **OPTIMASI PENJADWALAN UJIAN OTOMATIS DENGAN MENGGUNAKAN ALGORITMA GREEDY – SIMULATED ANNEALING – HYPER HEURISTIC**

yang merupakan salah satu syarat kelulusan pada Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya

Penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu dalam pengerjaan Tugas Akhir ini, khususnya kepada :

1. Seluruh anggota keluarga khususnya alm. Papa Ahmad Suchani dan mama Wiwik Sulianingsih yang selalu memberikan dukungan baik secara moril maupun materil demi tercapainya Tugas Akhir ini.
2. Bapak Ahmad Muklason, S.Kom., M.Sc., Ph.D. selaku dosen pembimbing yang meluangkan waktu, memberikan ilmu, petunjuk, dan motivasi untuk kelancaran Tugas Akhir ini.
3. Ibu Feby Artwodini Muqtadiroh S.Kom, M.T. selaku dosen wali, terima kasih atas bimbingan yang diberikan selama penulis menjadi mahasiswa sarjana di Jurusan Sistem Informasi ITS.
4. Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu yang sangat berharga bagi penulis.
5. Sahabat yang selalu memberikan dukungan dan motivasi selama perkuliahan Dilo, Faza, Rijal, Tasya, Yoga
6. Teman satu tim Gigih dan Putri yang sudah sangat membantu penulis dalam pengerjaan tugas akhir ini.

7. Seluruh keluarga FOXIS yang selalu memberikan dukungan serta doa.

Dan berbagai pihak yang tidak dapat penulis tuliskan namanya satu per satu. Semoga Tuhan membalas semua kebaikan yang telah dilakukan. Penulis menyadari banyaknya kekurangan yang dapat dikembangkan pada tugas akhir ini, oleh karena itu penulis menerima saran dan kritik yang membangun. Semoga tugas akhir ini dapat memberikan manfaat.

Surabaya, Januari 2018

Penulis



## DAFTAR ISI

ABSTRAK .....	ix
ABSTRACT .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xix
DAFTAR TABEL .....	xxi
DAFTAR SEGMENT KODE PROGRAM .....	xxiii
BAB I PENDAHULUAN .....	1
2.1. Latar Belakang .....	1
2.2. Perumusan Masalah .....	2
2.3. Batasan Pengerjaan Tugas Akhir .....	2
2.4. Tujuan Tugas Akhir .....	3
2.5. Manfaat Tugas Akhir .....	3
2.6. Sistematika Penulisan .....	4
2.7. Relevansi .....	5
BAB II TINJAUAN PUSTAKA .....	7
3.1. Penjadwalan .....	7
3.2. Carter Dataset .....	8
3.3. Pendekatan State-of-the-art .....	9
3.4. Algoritma Hyper-heuristics .....	10
3.5. Algoritma <i>Greedy</i> .....	11
3.6. Algoritma <i>Simulated Annealing</i> .....	12
BAB III METODOLOGI .....	17
4.1. Identifikasi Masalah .....	17
4.2. Penyiapan Data .....	17
4.3. Praproses Data .....	19
4.4. Formulasi Fungsi Tujuan .....	19
4.5. Implementasi .....	19
4.5.1. Implementasi Algoritma <i>Greedy</i> .....	19
4.5.2. Implementasi Algoritma <i>Simulated Annealing</i> .....	20
4.6. Uji Coba dan Analisis Jadwal Ujian .....	21
4.7. Penyusunan Buku Tugas Akhir .....	21
BAB IV PERANCANGAN .....	23

5.1.	Pengumpulan dan Deskripsi Data.....	23
5.1.1.	Deskripsi Data Mahasiswa.....	23
5.1.2.	Deskripsi Data Mata Kuliah.....	25
5.2.	Pra-Proses Data.....	26
5.3.	Formulasi Fungsi Tujuan .....	29
5.3.1.	Variabel Keputusan.....	29
5.3.2.	Model Matematis <i>Hard Constraints</i> .....	30
5.3.3.	Fungsi Tujuan .....	31
5.3.4.	Pembentukan Matriks Jumlah mahasiswa .....	31
5.3.5.	Pembentukan Matriks Selisih Slot .....	32
5.4.	Implementasi Algoritma Greedy .....	33
5.5.	Implementasi Algoritma <i>Simulated Annealing</i> .....	33
BAB V IMPLEMENTASI.....		35
6.1.	Membaca File Data Input .....	35
6.2.	Pembuatan Matriks Konflik.....	36
6.3.	Penerapan Algoritma Greedy.....	37
6.4.	Penerapan Algoritma Simulated Annealing.....	38
BAB VI HASIL DAN PEMBAHASAN .....		43
7.1.	Data Uji Coba .....	43
7.2.	Lingkungan Uji Coba .....	43
7.3.	Fungsi tujuan dari jadwal manual .....	44
7.4.	Hasil Uji Coba Algoritma <i>Greedy</i> .....	46
7.4.1.	Penjadwalan UTS.....	46
7.4.2.	Penjadwalan UAS .....	49
7.5.	Hasil Uji Coba Algoritma Simulated Annealing ....	51
7.5.1.	Penjadwalan UTS.....	51
7.5.2.	Penjadwalan UAS .....	54
7.6.	Hasil dan Pembahasan Uji Coba.....	57
7.6.1.	Skenario 1 : Perubahan Iterasi Setiap Temperatur .....	58
7.6.2.	Skenario 2 : Pemilihan <i>Low Level Heuristic</i> ...	63
7.6.3.	Skenario 3 : Perubahan Koefisien Pendinginan.....	68
7.6.4.	Pembahasan Hasil Skenario Uji Coba.....	73
7.7.	Perbandingan dengan Algoritma Lain .....	73
BAB VII KESIMPULAN DAN SARAN .....		77
8.1.	Kesimpulan .....	77

8.2. Saran .....	77
DAFTAR PUSTAKA .....	79
RIWAYAT PENULIS .....	83
LAMPIRAN 1 .....	11-1
LAMPIRAN 2 .....	12-1
LAMPIRAN 3 .....	13-1
LAMPIRAN 4 .....	14-1

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2.1 Framework algoritma hyper-heuristics.....	10
Gambar 2.2 Ilustrasi Perbandingan Algoritma.....	15
Gambar 3.1. Metode Pengerjaan Tugas Akhir .....	18
Gambar 6.1 Permasalahan pada Jadwal UTS Manual .....	44
Gambar 6.2 Permasalahan pada Jadwal UAS Manual .....	45
Gambar 6.3 Grafik Fungsi Tujuan dari SA .....	52
Gambar 6.4 Jadwal 12 Slot yang bertabrakan .....	55
Gambar 6.5 Fungsi Tujuan dengan Iterasi 3000 .....	58
Gambar 6.6 Fungsi Tujuan dengan Iterasi 5000 .....	59
Gambar 6.7 Fungsi Tujuan dengan Iterasi 7000 .....	60
Gambar 6.8 Fungsi Tujuan dengan Iterasi 100000 .....	61
Gambar 6.9 Box-plot Nilai Fungsi Uji Coba .....	62
Gambar 6.10 Pemilihan Low Level Heuristic.....	63
Gambar 6.11 Fungsi Tujuan <i>move</i> .....	64
Gambar 6.12 Fungsi Tujuan <i>swap</i> .....	65
Gambar 6.13 Fungsi Tujuan Acak .....	66
Gambar 6.14 Box-plot Nilai Fungsi Uji Coba .....	67
Gambar 6.15 Fungsi Tujuan Koefisien 0,9 .....	69
Gambar 6.16 Fungsi Tujuan Koefisien 0,85 .....	70
Gambar 6.17 Fungsi Tujuan Koefisien 0,5 .....	71
Gambar 6.18 Box-plot Uji Coba Perubahan Koefisien Penurunan Temperatur .....	72
Gambar 6.19 Grafik Perbandingan Selama 1 Menit .....	74
Gambar 6.20 Grafik Perbandingan Selama 5 Menit .....	75

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2.1 Carter Dataset.....	8
Tabel 4.1 Ringkasan Data .....	23
Tabel 4.2 Potongan Data Mahasiswa .....	23
Tabel 4.3 Data Mata Kuliah .....	25
Tabel 4.4 Konversi Kode Mata Kuliah .....	27
Tabel 4.5 Potongan file .crs.....	28
Tabel 4.6 Potongan Hasil Pivot Table Data Mahasiswa .....	28
Tabel 4.7 Potongan Data Mahasiswa .....	29
Tabel 4.8 Potongan Matriks Konflik.....	32
Tabel 4.9 Potongan Matrik Konflik Selisish Slot.....	32
Tabel 6.1 Spesifikasi Perangkat Keras .....	43
Tabel 6.2 Spesifikasi Perangkat Lunak .....	43
Tabel 6.3 Mata Kuliah yang Bertabrakan .....	45
Tabel 6.4 Algoritma <i>Greedy</i> Gagal Menjadwalkan .....	46
Tabel 6.5 Perbandingan Hasi Algoritma Greedy dengan Jadwal Manual.....	47
Tabel 6.6 Perbandingan Jadwal Hasil Algoritma <i>Greedy</i> .....	49
Tabel 6.7 Parameter Simulated Annealing.....	51
Tabel 6.8 Perbandingan Hasil Jadwal Ujian .....	52
Tabel 6.9 Matakuliah yang Dioptimasi .....	54
Tabel 6.10 Jadwal UAS 12 Slot Usulan SA.....	55
Tabel 6.11 Jadwal UAS 15 Slot Usulan SA.....	56
Tabel 6.12 Hasil Uji Coba Perubahan Iterasi .....	61
Tabel 6.13 Perbandingan Jadwal dari Perubahan Iterasi.....	62
Tabel 6.14 Perbandingan Hasil Pemilihan LLH .....	66
Tabel 6.15 Perbandingan Jadwal dari Perubahan Metode LLH .....	68
Tabel 6.16 Perbandingan Hasil Uji Coba Perubahan Koefisien Penurunan Temperatur .....	71
Tabel 6.17 Perbandingan Jadwal hasil Uji Coba.....	72
Tabel 10.1 Pemetaan Mahasiswa Terhadap Matakuliah .....	11-1
Tabel 12.1 Usulan Jadwal UTS Hasil Simulated Annealing.....	13-

Tabel 13.1 Usulan Jadwal UAS Hasil Simulated Annealing 14-1



## DAFTAR SEGMENT KODE PROGRAM

Kode 2.1 Pseudocode untuk Simulated Annealing .....	13
Kode 5.1 Method Baca File crs .....	35
Kode 5.2 Method Baca File stu .....	36
Kode 5.3 Method Konflik Matriks .....	37
Kode 5.4 Method okToSlot .....	37
Kode 5.5 Method okToRoom .....	38
Kode 5.6 Method untuk Menentukan Slot Ujian .....	38
Kode 5.7 Inisialisasi Variabel untuk Simulated Annealing ...	39
Kode 5.8 Kode untuk Mengubah Slot .....	39
Kode 5.9 Method Move .....	39
Kode 5.10 Method Swap .....	40
Kode 5.11 Simulated Annealing .....	41

*Halaman ini sengaja dikosongkan*

# **BAB I**

## **PENDAHULUAN**

Pada bab ini akan dibahas mengenai pendahuluan tugas akhir yang berisi latar belakang, perumusan masalah, batasan pengerjaan tugas akhir, tujuan dan manfaat dari pengerjaan tugas akhir serta sistematika penulisan buku tugas akhir.

### **2.1. Latar Belakang**

Penyusunan jadwal atau penjadwalan diperlukan untuk mengatur waktu, sehingga didapatkan jadwal yang seefisien mungkin. Suatu penjadwalan akan tampak sederhana apabila komponen yang akan dijadwalkan dalam jumlah kecil, namun penjadwalan akan menjadi masalah rumit dan memakan waktu yang tidak sebentar apabila komponen penyusunnya terdapat dalam jumlah besar.

Dalam bidang pendidikan, ujian adalah suatu kegiatan yang sangat penting. Melalui ujian pihak lembaga pendidikan dapat mengukur sejauh mana pemahaman peserta didik dalam memahami materi yang telah diajarkan. Untuk dapat melaksanakan kegiatan ujian, pihak lembaga pendidikan perlu melakukan penjadwalan ujian. Pada tingkat perguruan tinggi, dimana setiap mahasiswanya berhak mengambil mata kuliah yang diinginkan, proses penjadwalan menjadi sebuah kegiatan yang lebih sulit. Dikarenakan penjadwalan ujian selama ini masih dilakukan secara manual, pihak perguruan tinggi harus memastikan bahwa jadwal ujian yang dihasilkan untuk masing-masing mahasiswa untuk suatu mata kuliah tidak bertabrakan dengan mata kuliah lain yang lain dalam satu slot waktu ujian. Selain itu proses penjadwalan yang dilakukan masih secara manual menyebabkan proses penjadwalan ujian membutuhkan waktu yang tidak sebentar.

Terdapat beberapa metode atau pendekatan yang dapat digunakan untuk menyelesaikan permasalahan penjadwalan. Pendekatan ini dikenal sebagai state-of-the-art approach. Salah satu pendekatan yang ada adalah hyperheuristics. Salah satu

keunggulan dari hyperheuristics adalah tidak dibutuhkannya parameter tuning untuk set permasalahan penjadwalan yang berbeda.

Dengan penggunaan pendekatan hyperheuristics diharapkan adanya sebuah sistem penjadwalan ujian yang mampu menghasilkan jadwal ujian yang lebih optimal dan membutuhkan waktu yang lebih singkat.

## **2.2. Perumusan Masalah**

Permasalahan yang diangkat dalam tugas akhir ini adalah sebagai berikut:

- a. Bagaimana merumuskan permasalahan penjadwalan ujian dalam bentuk model matematis.
- b. Bagaimana menerapkan algoritma greedy simulated annealing hyper-heuristic pada model matematis.
- c. Bagaimana kinerja sistem optimalisasi yang menerapkan algoritma greedy simulated annealing hyper-heuristic.
- d. Bagaimana perbandingan kinerja antara sistem yang menerapkan algoritma *greedy simulated annealing hyper-heuristic* dengan penjadwalan yang dilakukan dengan cara manual.

## **2.3. Batasan Pengerjaan Tugas Akhir**

Batasan yang ada dalam pengerjaan tugas akhir ini meliputi:

- a. Data yang akan digunakan adalah data mahasiswa semester genap tahun ajaran 2016-2017 Jurusan Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya.
- b. Hasil dari tugas akhir ini ada sistem penjadwalan ujian otomatis berbasis Java dan dibangun menggunakan Netbeans IDE 8.2
- c. Penjadwalan yang dilakukan hanya untuk penjadwalan waktu ujian mahasiswa.

- d. Kapasitas ruang ujian akan digunakan sebagai *hard constraints*.
- e. Matakuliah hanya dijadwalkan pada satu slot waktu

## 2.4. Tujuan Tugas Akhir

Tujuan yang ingin dicapai pada pengerjaan tugas akhir ini adalah membuat suatu sistem optimalisasi penjadwalan ujian pada Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya dengan menerapkan algoritma *greedy simulated hyper heuristic*. Diharapkan sistem ini dapat memperbaiki sistem penjadwalan yang sudah ada dan mampu menghasilkan jadwal ujian yang lebih optimal untuk mahasiswa.

## 2.5. Manfaat Tugas Akhir

Manfaat yang diberikan dengan adanya tugas akhir ini adalah sebagai berikut:

### Manfaat Teoritis

Adanya penelitian ini diharapkan dapat menambah khasanah keilmuan dalam bidang sistem informasi khususnya pada bidang penjadwalan, hal ini dikarenakan algoritma *greedy simulated annealing hyper-heuristic* merupakan algoritma baru yang masih belum banyak digunakan..

### Manfaat Praktis

- a. Adanya penelitian ini diharapkan dapat membantu pihak Jurusan Sistem Informasi dalam penyusunan jadwal ujian baik itu tengah semester maupun akhir semester, baik itu dari segi waktu yang dibutuhkan dalam penyusunan jadwal maupun mengurangi kemungkinan terjadinya jadwal ujian yang bentrok.
- b. Adanya penelitian ini diharapkan jadwal ujian mahasiswa dapatkan nantinya menjadi lebih optimal sehingga secara tidak langsung membantu mahasiswa memperoleh hasil ujian yang lebih baik.

## **2.6. Sistematika Penulisan**

Buku tugas akhir ini disusun berdasarkan urutan pelaksanaan penelitian yang mana antara bab satu dengan yang lainnya saling berhubungan. Buku tugas akhir ini disusun menjadi 7 bab dengan susunan sebagai berikut.

### **BAB I PENDAHULUAN**

Bab ini menjelaskan hal-hal dasar mengenai isi tugas akhir ini. Hal-hal yang dijelaskan di sini antara lain: latar belakang pengerjaan tugas akhir, rumusan permasalahan pengerjaan tugas akhir, batasan permasalahan pengerjaan tugas akhir, tujuan pengerjaan tugas akhir, manfaat dari pengerjaan tugas akhir, dan sistematika penulisan tugas akhir

### **BAB II TINJAUAN PUSTAKA**

Pada bab ini dijelaskan mengenai teori-teori terkait yang bersumber dari buku, jurnal, ataupun artikel yang berfungsi sebagai dasar dalam melakukan pengerjaan tugas akhir agar dapat memahami konsep atau teori penyelesaian permasalahan yang ada.

### **BAB III METODOLOGI Pengerjaan Tugas Akhir**

Pada bab ini dijelaskan mengenai langkah–langkah pengerjaan tugas akhir beserta metode yang digunakan. Langkah–langkah pengerjaan dijelaskan dalam sebuah diagram alur yang sistematis dan akan dijelaskan tahap demi tahap

### **BAB IV PERANCANGAN APLIKASI**

Pada bab ini dijelaskan mengenai rancangan aplikasi yang akan dibangun pada tugas akhir ini

### **BAB V IMPLEMENTASI APLIKASI**

Pada bab ini akan dipaparkan mengenai cara pembangunan aplikasi berdasarkan desain aplikasi yang telah dibuat sebelumnya. Aplikasi yang dibangun menggunakan bahasa pemrograman JAVA.

### **BAB VI UJI COBA DAN ANALISIS HASIL**

Pada bab ini akan dilakukan uji coba terhadap aplikasi yang terbagi menjadi dua skenario uji coba, yakni uji verifikasi dan uji validasi. Selain itu, juga dijelaskan mengenai analisis terhadap hasil keluaran aplikasi.

## **BAB VII PENUTUP**

Bab ini berisi kesimpulan dari pengerjaan tugas akhir dan saran untuk pengembangan lebih lanjut.

### **2.7. Relevansi**

Topik tugas akhir ini yaitu mengenai optimasi penjadwalan ujian. Topik tersebut sesuai dengan bidang ilmu riset operasi yang menjadi cakupan research roadmap pada laboratorium Rekayasa Data dan Inteligensi Bisnis (RDIB).

*Halaman ini sengaja dikosongkan*



## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan dibahas mengenai tinjauan pustaka dari tugas akhir. Bab ini berisi dasar teori yang mendukung tugas akhir sehingga ada dasar teori yang cukup kuat sebagai pendukung pelaksanaan tugas akhir.

#### **3.1. Penjadwalan**

Penjadwalan adalah suatu kajian matematis yang bertujuan untuk mencari solusi yang optimal pada penyusunan, pengelompokan, pengurutan atau pemilihan objek diskrit yang biasanya jumlahnya terbatas (finite number) [1]. Sedangkan pada penelitian yang lain [2] menyatakan bahwa penjadwalan adalah perencanaan yang berbasiskan waktu dan combinational optimization yang diselesaikan dengan menggunakan pencarian dan heuristik yang biasanya mengarah kepada solusi yang memenuhi namun suboptimal. Dalam kehidupan sehari-hari, masalah penjadwalan dapat muncul pada bidang pendidikan seperti pada ujian maupun jadwal perkuliahan ataupun pada bidang lain seperti transportasi

Secara khusus pada penelitian [3] mendefinisikan permasalahan penjadwalan di bidang pendidikan sebagai permasalahan yang memiliki empat parameter himpunan terbatas (finite set), yaitu timeslot atau slot waktu (T), sumber daya (S) yang berupa ruang kelas, jumlah pertemuan (P) dan batasan atau constraints (B). Penjadwalan dilakukan untuk mengalokasikan slot waktu dan sumber daya pada pertemuan dengan memenuhi batasan semaksimal mungkin. Pada penelitian lain [4] menjelaskan bahwa batasan (constraints) dibagi menjadi dua macam, yaitu hard constraints dan soft constraints. Hard constraints didefinisikan sebagai batasan yang harus dipenuhi, antara lain:

- a. Seluruh matakuliah harus dijadwalkan
- b. Tidak terjadi bentrok jadwal

- c. Jumlah peserta ujian yang hadir pada satu sesi atau slot tidak boleh melebihi kapasitas total dari kelas

Sedangkan *soft constraints* didefinisikan sebagai batasan yang dapat dilanggar namun harus diminimalisir jumlahnya, batasan ini antara lain:

- a. Ada ujian pada slot waktu terakhir
- b. Seorang mahasiswa memiliki lebih dari satu ujian dalam sehari
- c. Seorang mahasiswa memiliki dua ujian berturut-turut

### 3.2. Carter Dataset

*Carter dataset* atau yang juga dikenal sebagai dataset Universitas Toronto adalah sebuah dataset yang terdiri dari 13 kasus dunia nyata yang dapat digunakan untuk permasalahan penjadwalan ujian [5]. Ringkasan *carter dataset* dapat dilihat pada tabel 2.1

**Tabel 3.1 Carter Dataset**

Data	Jumlah Ujian	Jumlah Siswa	Jumlah Slot
CAR91	682	16925	35
CAR92	543	18419	32
EAR83	190	1125	24
HEC92	81	2823	18
KFU93	461	5349	20
LSE91	381	2726	18
PUR93	2419	30029	42
RYE92	486	11483	23
STA83	139	611	13
TRE92	261	4360	23
UTA92	622	21266	35
UTE92	184	2749	10
YOR83	181	941	21

Dataset ini kemudian menjadi standart untuk permasalahan penjadwalan ujian.

### 3.3. Pendekatan State-of-the-art

Pendekatan state of the art merupakan pendekatan yang digunakan dalam penyelesaian penjadwalan menggunakan metode-metode tertentu [5]. Metode-metode tersebut antara lain:

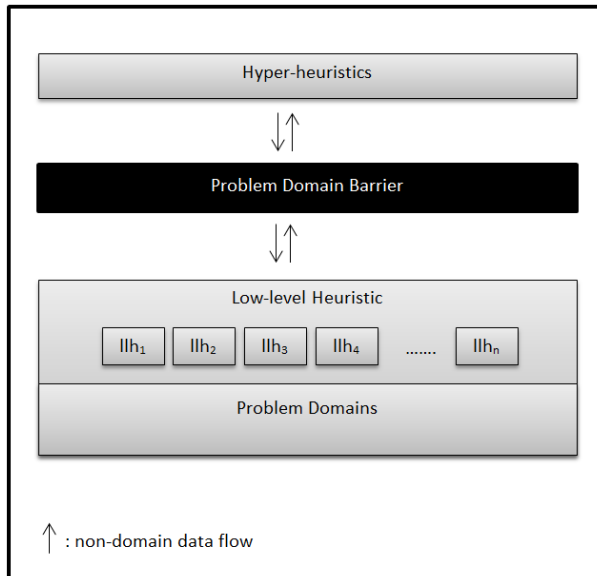
- a. *Graph based sequential*  
 Pada graph based sequential penjadwalan dilakukan dengan membangun solusi yang berawal dari set kosong yang kemudian masing-masing slot waktu ujian diisikan satu per satu, sehingga tidak ada constraints yang dilanggar pada setiap slot waktu. Graph based sequential terdiri dari direct heuristic, exact vertex colouring, dan graph colouring heuristic
- b. *Clustering*  
 Pada metode *clustering* setiap set mata kuliah yang memenuhi segala *hard* constraints di kelompokkan kedalam kelompok yang sama, kemudian masing-masing kelompok tersebut dimasukkan kedalam satu slot waktu untuk memenuhi *soft constraints*.
- c. *Metaheuristic*  
*Metaheuristic* merupakan salah satu metode penjadwalan yang paling banyak digunakan untuk penyelesaian masalah penjadwalan. Secara khusus metode *metaheuristic* digunakan untuk memperbaiki solusi yang pada awalnya telah dihasilkan oleh metode *sequential*. Kelemahan dari metode ini membutuhkan *parameter tuning* untuk setiap set permasalahan yang berbeda yang menyebabkan metode ini tidak generik
- d. *Multi-criteria*  
 Pada *multi-criteria*, setiap batasan yang ada didefinisikan melalui sudut pandang yang berbeda dan bukan hanya dianggap sebagai kumpulan dari *cost* dan *penalties* yang disebabkan oleh batasan yang berbeda

e. *Hyper-heuristic*

Metode *Hyper-heuristic* adalah metode generik yang digunakan untuk mengatasi keterbatasan yang ada pada metode *metaheuristic*, dimana metode *metaheuristic* membutuhkan *parameter tuning* untuk menyelesaikan setiap set masalah yang berbeda.

### 3.4. Algoritma Hyper-heuristics

Secara umum *hyper-heuristic* adalah sebuah kumpulan pendekatan yang memiliki tujuan untuk otomatisasi suatu proses yang menggunakan *machine learning* untuk memilih dan mengkombinasikan heuristics yang lebih sederhana atau menghasilkan heuristics baru dengan menggunakan komponen dari heuristics yang telah ada. Tujuan dari *hyper-heuristic* adalah untuk menghasilkan heuristic yang bersifat generik sehingga dapat digunakan pada set permasalahan yang berbeda.



**Gambar 3.1 Framework algoritma hyper-heuristics**

Pada metode *hyper-heuristic* pencarian dilakukan diatas *search space* yang berupa *low-level heuristics*, hal ini dikarenakan pada *hyper-heuristic* terdapat *problem domain barrier* yang mengakibatkan *hyper-heuristics* tidak langsung bersinggungan dengan *solution space* namun hanya bersinggungan dengan *low-level heuristic*. Dibandingkan dengan metode *meta-heuristics* yang bekerja diatas *search space* yang berupa *solution space*, metode *hyper-heuristic* tidak bergantung pada *problem domain* tertentu saja sehingga tidak diperlukan *parameter tuning* untuk setiap problem domain secara manual.

### 3.5. Algoritma Greedy

Algoritma *greedy* merupakan algoritma yang digunakan dalam penyelesaian masalah optimalisasi dengan cara memilih solusi yang paling optimal dengan harapan solusi tersebut memberikan solusi akhir yang juga optimal. Secara umum cara kerja dari algoritma *greedy* adalah mencari solusi terbaik yang dapat diperoleh pada suatu langkah tanpa memperhatikan solusi yang ada pada langkah berikutnya. Pada algoritma *greedy* terdapat lima komponen antara lain:

- a. Himpunan Kandidat  
Himpunan kandidat merupakan himpunan yang berisi elemen yang nantinya membentuk solusi.
- b. Fungsi Seleksi  
Fungsi seleksi memiliki tujuan untuk memilih kandidat terbaik untuk dimasukkan kedalam solusi sehingga solusi optimal dapat terbentuk.
- c. Fungsi Kelayakan  
Fungsi kelayakan bertujuan untuk memeriksa kandidat yang telah dipilih dapat menghasilkan solusi yang layak
- d. Fungsi Objektif  
Fungsi objektif bertujuan untuk memberikan nilai kepada solusi
- e. Himpunan Solusi  
Himpunan solusi berisi solusi yang dapat digunakan dalam penyelesaian masalah optimasi

Kelemahan dari algoritma greedy adalah solusi yang diberikan pada setiap langkah belum tentu menghasilkan solusi yang terbaik. Hal ini dikarenakan fokus dari algoritma greedy adalah menemukan solusi lokal yang optimal padahal solusi lokal tersebut belum tentu mengarah pada solusi global yang terbaik atau dikenal sebagai pseudo-optimum..

### **3.6. Algoritma *Simulated Annealing***

Algoritma simulated annealing adalah algoritma untuk menyelesaikan permasalahan optimisasi yang diadaptasi dari proses annealing logam pada bidang metalurgi yang melibatkan pemanasan dan pendinginan secara bertahap untuk menghasilkan efisiensi dalam penggunaan energi yang optimal [6]. Pada proses pengolahan, suatu logam akan dilakukan pemanasan hingga mencapai temperatur tertentu dimana pada saat itu partikel yang terdapat pada logam tersebut dapat bergerak bebas dengan tingkat energi yang tinggi. Kemudian, secara perlahan-lahan Temperatur diturunkan dengan harapan energi yang ada dapat berkurang. Semakin lambat laju pendinginan, maka semakin rendah pula energi yang akan dicapai. Hingga nantinya partikel logam tersebut berada pada posisi yang optimum dengan penggunaan energi yang minimum.

```

Set the maximum number of iteration  $I = I_{\max}$ 
Set the starting temperature  $T = T_0$ 
Set Cooling rate.
Generate the initial solution  $S = S_0$ 
Repeat
Repeat
Generate a random neighbour  $s^*$ 
 $\Delta E = f(s^*) - f(s)$ 
If  $\Delta E \leq 0$  Then  $s = s^*$ /accept the neighbour solution
Else Accept  $s^*$  with probability  $e^{\frac{-\Delta E}{T}}$ 
Until Equilibrium condition
Reduce temperature rate  $T = T * \text{Cooling rate}$ 
Update the best solution
Until Stopping criteria satisfied  $l \leq l_{\min}$ 

```

### Kode 3.1 Pseudocode untuk Simulated Annealing

Terdapat tiga komponen yang perlu diperhatikan dalam algoritma *simulated annealing* [7]. Komponen tersebut antara lain:

a. Proses annealing

Proses annealing adalah proses utama yang ada pada algoritma simulated annealing. Proses ini bertujuan agar sistem tidak berada pada keadaan lokal minimum. Proses ini bergantung pada parameter berikut:

1. Temperatur awal

Temperatur awal dipilih cukup tinggi untuk memperluas penerimaan terhadap solusi baru. Pada penelitian terdahulu [8], Temperatur awal dipilih sebesar 1 dikarenakan nilai kemungkinan maksimum. Sedangkan di penelitian lain [9] dan [10] menggunakan nilai sebesar 50% dari nilai fungsi tujuan awal

2. Jumlah iterasi pada setiap Temperatur

Penurunan Temperatur akan dilakukan apabila telah mencapai jumlah iterasi tertentu. Jumlah iterasi yang dilakukan pada setiap Temperatur dapat bernilai konstan atau berubah-ubah

### 3. Parameter penurunan temperatur

Penurunan temperatur dilakukan dengan menggunakan variable  $a$  yang disebut sebagai cooling rate dengan nilai antara 0 hingga 1. Pada penelitian terdahulu nilai  $a$  dicapai pada  $0,8 \leq a < 1$  [10]. Proses penurunan Temperatur pada algoritma *simulated annealing* dilakukan secara bertahap.

#### b. Penyusunan ulang

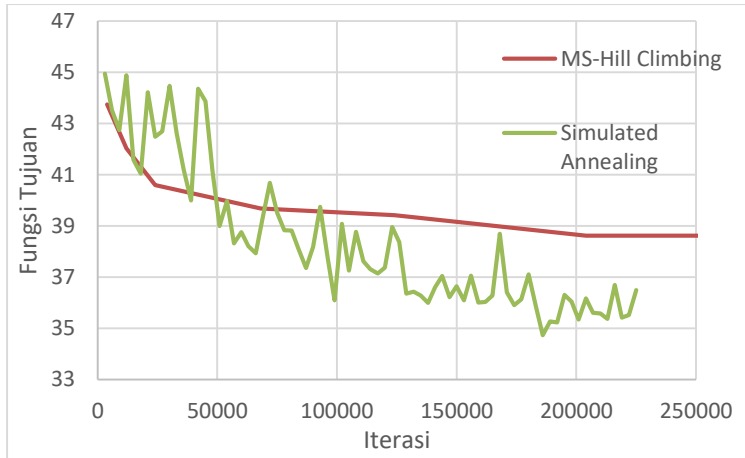
Penyusunan ulang atau pembentukan solusi lingkungan dilakukan dengan mengubah konfigurasi secara acak. Penyusunan ulang ini dapat dilakukan dengan berbagai cara bergantung pada permasalahan

#### c. Penghentian algoritma

Penghentian algoritma ditentukan sejak awal proses. Kriteria penghentian algoritma dapat berupa Temperatur minimum, dimana proses akan berhenti apabila telah mencapai Temperatur tertentu atau dapat juga dengan menggunakan durasi waktu, dimana proses akan berhenti apabila algoritma telah berjalan selama durasi tertentu telah terpenuhi walaupun belum mencapai solusi yang optimal

Kelebihan algoritma *simulated annealing* dibandingkan dengan algoritma seperti *hill climbing* adalah pada saat algoritma *hill climbing* hanya dapat menerima solusi baru yang lebih baik sehingga terdapat kemungkinan solusi akhir berada pada keadaan *local optimum*, algoritma *simulated annealing* dapat mengatasi hal ini dengan menerima solusi baru yang lebih buruk sehingga mampu mencari solusi diluar solusi *local optimum*.





**Gambar 3.2 Ilustrasi Perbandingan Algoritma**

*Halaman ini sengaja dikosongkan*

## **BAB III METODOLOGI**

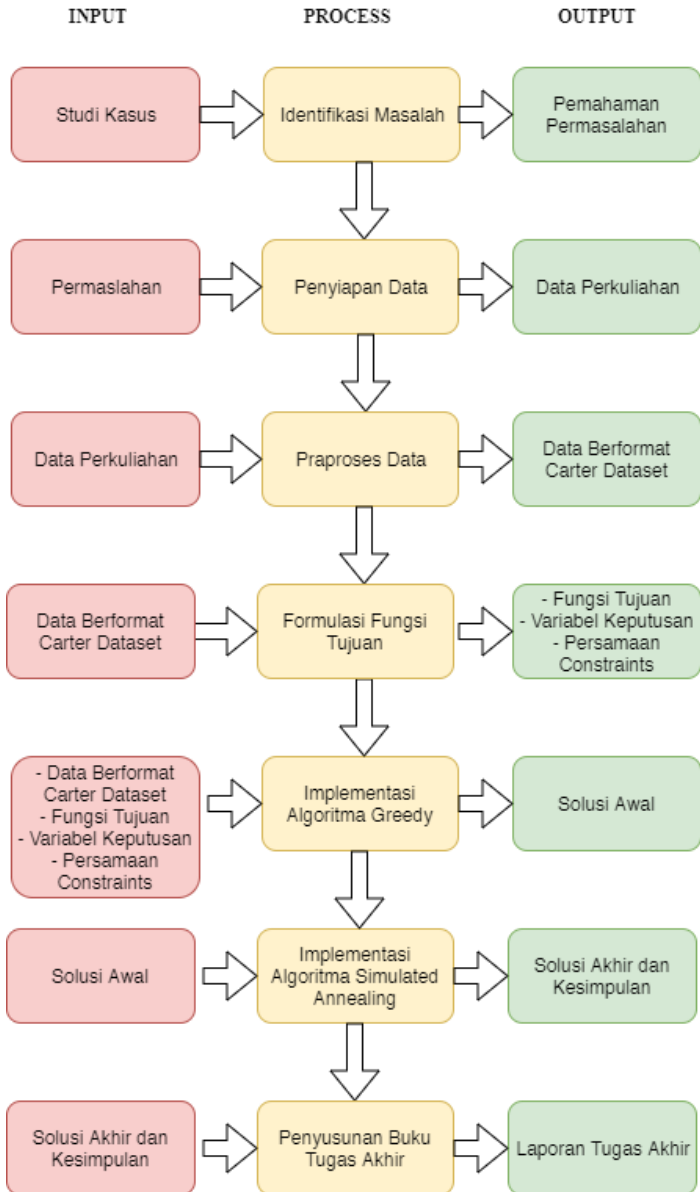
Pada bab ini akan dijelaskan mengenai alur metodologi yang akan dilakukan dalam tugas akhir ini. Metodologi ini juga digunakan sebagai pedoman untuk melaksanakan tugas akhir agar terarah dan sistematis. Gambar 3.1 menunjukkan metodologi tugas akhir yang digunakan.

### **4.1. Identifikasi Masalah**

Tahap identifikasi masalah adalah tahapan untuk memahami masalah yang hendak diselesaikan. Pada tahap ini dilakukan memahami permasalahan tentang penjadwalan ujian yang ada pada Departemen Sistem Informasi. Diketahui bahwa dalam pembuatan jadwal ujian, pihak Departemen Sistem Informasi masih melakukan dengan cara manual sehingga dalam penyusunan jadwal membutuhkan waktu yang tidak sebentar. Pada tahap ini penulis menetapkan tujuan yang hendak dicapai serta batasan dalam pengerjaan. Pada tahapan ini juga dilakukan studi literatur dari penelitian terdahulu untuk mengetahui pendekatan dan metode algoritma yang digunakan dalam penyelesaian permasalahan yang sejenis, selain itu juga berguna sebagai dasar teori dalam penyelesaian masalah. Dari tahapan ini diketahui bahwa permasalahan yang berupa optimalisasi jadwal ujian mahasiswa diwakili dengan nilai *proximity cost* sebagai fungsi tujuan.

### **4.2. Penyiapan Data**

Pada tahap ini dilakukan pengumpulan data yang dibutuhkan untuk pengerjaan tugas akhir ini. Data yang dikumpulkan berupa dataset peserta UTS dan UAS semester genap tahun ajaran 2016-2017 Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Sebagai batasan data yang digunakan adalah data untuk jenjang S1. Data yang telah didapatkan akan diolah pada tahap praproses



**Gambar 4.1. Metode Pengerjaan Tugas Akhir**

### 4.3. Praproses Data

Pada tahapan ini, data yang telah didapatkan kemudian dilakukan pra-proses yang bertujuan untuk menyesuaikan format data dengan format data pada *carter dataset* untuk digunakan dalam proses pemodelan. Data yang telah diproses dapat digunakan sebagai masukan algoritma *greedy*.

### 4.4. Formulasi Fungsi Tujuan

Pada tahap ini dilakukan pembuatan model awal untuk menyelesaikan permasalahan yang hendak diselesaikan berdasarkan variabel-variabel seperti

1. Variabel Keputusan  
Memastikan bahwa setiap matakuliah di tempatkan pada suatu slot ujian
2. Fungsi Tujuan  
Meminimalkan nilai fungsi tujuan berupa *proximity cost*
3. Fungsi Batasan (*Constraints*)  
Jadwal ujian yang dibuat diharuskan untuk memenuhi batasan sebagai berikut:
  - a. Seluruh matakuliah terjadwalkan
  - b. Tidak terdapat matakuliah yang bentrok
  - c. Tidak melebihi kapasitas total dalam satu slot

### 4.5. Implementasi

Pada tahap implementasi dilakukan perancangan serta pengembangan sistem penjadwalan ujian. Sistem ini dibuat dengan menggunakan algoritma *greedy simulated annealing hyper heuristics* dan bahasa pemrograman Java menggunakan *Netbeans IDE 8.2*.

#### 4.5.1. Implementasi Algoritma *Greedy*

Pada tahap implementasi ini diterapkan algoritma *greedy* untuk mengolah data yang telah dilakukan pra-proses. Pada tahap ini data matakuliah terlebih dahulu akan diurutkan berdasarkan jumlah mahasiswa peserta kemudian satu-persatu matakuliah

dimasukkan kedalam suatu slot waktu, apabila matakuliah tersebut telah memenuhi dua buah *hard constraint* yaitu tidak ada mahasiswa yang bertabrakan dalam satu slot waktu serta kapasitas peserta ujian tidak melebihi kapasitas total kelas pada saat itu. Apabila suatu matakuliah memenuhi kedua *hard constraint* tersebut maka algoritma akan beralih ke matakuliah berikutnya, namun apabila tidak memenuhi maka algoritma akan memindahkan matakuliah tersebut ke slot lain dan kembali akan dipastikan apakah telah memenuhi dua *hard constraints*. Matakuliah yang telah memenuhi dua *hard constraint* akan menjadi solusi awal yang akan dilakukan optimasi pada algoritma *simulated annealing*.

#### **4.5.2. Implementasi Algoritma *Simulated Annealing***

Dalam penerapan algoritma *simulated annealing* memerlukan beberapa parameter yang perlu ditentukan terlebih dahulu seperti parameter temperatur awal yang menjadi kondisi awal dari solusi, parameter Temperatur akhir yang menjadi parameter untuk penghentian algoritma serta parameter koefisien penurunan temperatur yang digunakan untuk menurunkan temperatur sistem hingga mencapai temperatur akhir dimana setiap penurunan temperatur sistem akan mencari solusi baru. Pada tahap ini jadwal awal yang menjadi solusi awal hasil algoritma *greedy* akan dioptimasi dengan cara mengubah slot ujian matakuliah. Perubahan slot ujian dilakukan dengan menukar secara acak dua slot matakuliah dan mengganti salah satu slot ujian matakuliah dengan slot acak untuk menemukan solusi yang baru. Solusi baru ini kemudian akan dipastikan apakah telah memenuhi dua *hardconstraint*. Apabila solusi baru telah memenuhi dua *hardconstraint* maka sistem akan menurunkan temperatur sesuai dengan koefisien penurunan temperatur yang telah ditentukan dan kemudian mencari solusi baru hingga temperatur sistem mencapai temperatur akhir, namun apabila solusi baru tidak berhasil memenuhi salah satu atau kedua *hardconstraint* maka sistem akan kembali melakukan perubahan terhadap solusi.

#### **4.6. Uji Coba dan Analisis Jadwal Ujian**

Pada tahap ini, jadwal ujian yang telah dihasilkan oleh sistem akan dibandingkan dengan jadwal ujian yang dihasilkan secara manual. Selain itu juga dilakukan perbandingan jadwal ujian dengan jadwal ujian yang juga dihasilkan oleh sistem namun dengan nilai parameter yang berbeda untuk mengetahui pengaruh perubahan parameter terhadap fungsi tujuan, serta dilakukan perbandingan dengan metode optimasi lain untuk mengetahui perbandingan kinerja algoritma dalam jangka waktu tertentu.

#### **4.7. Penyusunan Buku Tugas Akhir**

Tahap penyusunan buku tugas akhir adalah proses pendokumentasian hasil tugas akhir serta analisis terhadap hasil akhir yang didapatkan. Keluaran dari tahap ini adalah buku tugas akhir. Buku tugas akhir ini diharapkan dapat menjadi referensi bagi penelitian selanjutnya.

*Halaman ini sengaja dikosongkan*



## **BAB IV PERANCANGAN**

Pada bab ini dijelaskan mengenai tahap perancangan dari penelitian tugas akhir serta bagaimana tugas akhir akan dilakukan

### **5.1. Pengumpulan dan Deskripsi Data**

Data yang digunakan dalam pengerjaan tugas akhir ini adalah data peserta didik dan mata kuliah yang diambil oleh masing-masing peserta didik Departemen Sistem Informasi tahun ajaran 2016-2017 semester genap. Ringkasan data yang didapatkan terdapat pada tabel 4.1

**Tabel 5.1 Ringkasan Data**

<b>Jumlah Matakuliah</b>	32
<b>Jumlah Mahasiswa</b>	567

#### **5.1.1. Deskripsi Data Mahasiswa**

Data mahasiswa sebagai peserta didik aktif Departemen Sistem Informasi pada semester genap tahun ajaran 2016-2017. Dari data yang didapatkan diketahui bahwa jumlah total peserta didik aktif adalah sebesar 567 mahasiswa. Data mahasiswa yang didapatkan adalah nrp, nama, kelas mata kuliah dan kode mata kuliah seperti yang terdapat pada tabel 4.2

**Tabel 5.2 Potongan Data Mahasiswa**

<b>SEM</b>	<b>NRP</b>	<b>NAMA</b>	<b>KELAS</b>	<b>KODE_MK</b>
2016,2	5215100121	KHANSA AL-FAIZIY	A	KS141205
2016,2	5215100134	AMANDA PERMATA IMAMAT RAJANI	A	KS141205
2016,2	5216100001	ROFIQOH NADIAH	A	KS141205
2016,2	5216100002	VITA AYU APRILLIA	A	KS141205

SEM	NRP	NAMA	KELAS	KODE_MK
2016,2	5216100003	ADITYA BUDI LAKSONO	A	KS141205
2016,2	5216100004	DINDA	A	KS141205
2016,2	5216100005	BAGUS TRI HANDOKO	A	KS141205
2016,2	5216100027	SAIFUDDIN NURUL MUHTI	A	KS141205
2016,2	5216100028	LULU'UL WATEF	A	KS141205
2016,2	5216100029	MUHAMMAD WILDAN MAULIDANI	A	KS141205
2016,2	5216100030	DINDA SULASTARI PRATIWI	A	KS141205
2016,2	5216100031	JANIS RAMADHANTI SAPUTRI	A	KS141205
2016,2	5216100052	PAWITRA GUSTI UPASADDHA	A	KS141205
2016,2	5216100053	YARDENI YULE SINURAT	A	KS141205
2016,2	5216100054	FIOSANDI RIZKY NOVALINO	A	KS141205
2016,2	5216100055	SALISAH ROSYIDAH R	A	KS141205
2016,2	5216100056	ETIKA PAHANG KRISDIYAN	A	KS141205
2016,2	5216100073	ADHI YOGA MURIS PAMUNGKAS	A	KS141205
2016,2	5216100075	SYANANTA PUTRA RAMADHAN	A	KS141205
2016,2	5216100077	AHMAD RIZQI FADLIL	A	KS141205

SEM	NRP	NAMA	KELAS	KODE_MK
2016,2	5216100078	MUHAMAD FAHAR CAHYANTO	A	KS141205
2016,2	5216100079	SATRYA BAGUS NURAWAN	A	KS141205
2016,2	5216100080	MUHAMMAD REZA PAHLAWAN	A	KS141205
2016,2	5216100081	AGUSTINA PANCA RINI	A	KS141205

### 5.1.2. Deskripsi Data Mata Kuliah

Data mata kuliah berisi jumlah peserta didik dari masing-masing mata kuliah yang diajarkan di Departemen Sistem Informasi pada tahun ajaran 2016-2017 semester genap. Adapun beberapa mata kuliah semester gasal juga dibuka pada semester genap dikarenakan mata kuliah tersebut menjadi prasyarat untuk mata kuliah di semester berikutnya. Pada semester tersebut diketahui pihak Departemen Sistem Informasi membuka sebanyak 32 mata kuliah dengan rincian yang terdapat pada tabel 4.3

**Tabel 5.3 Data Mata Kuliah**

nama_mk	CountOfnrp	Sem
Algoritma dan Struktur Data	155	2
Analisis dan Desain Perangkat Lunak	160	4
Arsitektur SI/TI Perusahaan	154	2
Bahasa Pemrograman	4	1
Dasar-Dasar Pengembangan Perangkat Lunak	28	3
Desain Basis Data	159	4
Desain dan Manajemen Proses Bisnis	17	3
e-Business	50	8
Etika Profesi SI	30	7
Forensika Digital	38	7

<b>nama_mk</b>	<b>CountOfnrp</b>	<b>Sem</b>
Interaksi Manusia dan Komputer	150	4
Keamanan Aset Informasi	161	4
Kepemimpinan dan Ketrampilan Interpersonal	152	2
Manajemen dan Administrasi Basis Data	184	6
Manajemen Kualitas SI/TI	40	8
Manajemen Layanan TI	21	5
Manajemen Pengadaan dan Investasi TI	138	6
Manajemen Rantai Pasok dan Hubungan Pelanggan	172	6
Manajemen Resiko TI	176	6
Pemrograman Berbasis Web	28	4
Pemrograman Integratif	18	8
Pemrograman Perangkat Bergerak	4	7
Pengantar Basis Data	1	3
Pengantar Sistem Operasi	150	2
Penggalian Data dan Analitika Bisnis	27	8
Pengukuran Kinerja dan Evaluasi TI	28	7
Riset Operasi Lanjut	28	8
Sistem Cerdas	191	6
Statistika	64	3
Tata Kelola TI	190	6
Tata Tulis Ilmiah	157	6
Teknik Peramalan	40	8

## **5.2. Pra-Proses Data**

Data yang baru didapatkan belum dapat langsung digunakan dalam sistem, sehingga membutuhkan pra-proses untuk mengubah format data sehingga sesuai dengan format data

*carter dataset*. Pra-proses data ini dilakukan dengan mengubah format kode mata kuliah dan nrp sehingga lebih sederhana. Konversi kode matakuliah dapat dilihat pada tabel 4.4

**Tabel 5.4 Konversi Kode Mata Kuliah**

<b>KODE_MK</b>	<b>Kode Convert</b>
KS141204	0001
KS141205	0002
KS141206	0003
KS141207	0004
KS141208	0005
KS141209	0006
KS141210	0007
KS141301	0008
KS141303	0009
KS141304	0010
KS141306	0011
KS141307	0012
KS141308	0013
KS141309	0014
KS141310	0015
KS141316	0016
KS141317	0017
KS141318	0018
KS141319	0019
KS141320	0020
KS141321	0021
KS141322	0022
KS141323	0023
KS141325	0024
KS141402	0025

KODE_MK	Kode Convert
KS141403	0026
KS141404	0027
KS141405	0028
KS141407	0029
KS141409	0030
KS141411	0031
KS141413	0032

Setelah dilakukan konversi, data matakuliah serta jumlah mahasiswa untuk masing-masing matakuliah tersebut disimpan dalam file dengan format .crs. Potongan dari file .crs dapat dilihat pada tabel 4.5

**Tabel 5.5 Potongan file .crs**

1	0001 4
2	0002 155
3	0003 150
4	0004 154
5	0005 152
6	0006 64
7	0007 157
8	0008 17
9	0009 28
10	0010 1
11	0011 160
12	0012 150
13	0013 161
14	0014 159
15	0015 144

Sedangkan untuk data mahasiswa, tahapan praproses dilakukan dengan mencari mata kuliah apa saja yang diambil setiap mahasiswa. Untuk melakukannya digunakan fungsi *pivot table* yang terdapat pada *Microsoft Excel* sehingga menghasilkan tabel seperti tabel 4.6

**Tabel 5.6 Potongan Hasil Pivot Table Data Mahasiswa**

	0001	0002	0003	0004	0005
5216100155	1	1	1	1	1

	0001	0002	0003	0004	0005
5216100156		1	1	1	1
5216100157		1	1	1	1
5216100159		1	1	1	1
5216100160		1	1	1	1
5216100162		1	1	1	1
5216100701		1	1	1	1
5216100702		1	1	1	1
5216100703		1	1	1	1
5216100704		1	1	1	1

Angka satu pada tabel 4.4 bermakna bahwa mahasiswa tersebut mengikuti mata kuliah tersebut. Setelah itu angka satu tersebut diganti dengan kode mata kuliah dan nrp mahasiswa diwakili dengan nomor baris, kemudian disimpan dalam file dengan ekstensi .stu. Potongan data mahasiswa pada file .stu dapat dilihat pada tabel 4.7

**Tabel 5.7 Potongan Data Mahasiswa**

558	0001	0002	0003	0004	0005
559	0002	0003	0004	0005	0009
560	0002	0003	0004	0005	
561	0002	0003	0004	0005	
562	0002	0003	0004	0005	0006
563	0002	0003	0004	0005	0006
564	0002	0003	0004	0005	
565	0002	0003	0004	0005	0006
566	0002	0003	0004	0005	0006
567	0002	0003	0004	0005	

### 5.3. Formulasi Fungsi Tujuan

#### 5.3.1. Variabel Keputusan

Variabel keputusan yang digunakan untuk memenuhi *hardconstraint* yang telah ditentukan sebelumnya, dimana variabel keputusan ini digunakan untuk memastikan bahwa

setiap matakuliah untuk dijadwalkan. dengan menggunakan persamaan matematis (1)

$$X_{it} \begin{cases} 1, & \text{jika mata kuliah ke } i \text{ dijadwalkan} \\ & \text{pada timeslot ke } t \\ 0, & \text{jika tidak} \end{cases} \quad (1)$$

Dimana,

$i = \{1, 2, 3, \dots, n\}$  adalah urutan matakuliah

$t = \{1, 2, 3, \dots, k\}$  adalah urutan sesi atau slot ujian yang tersedia

### 5.3.2. Model Matematis *Hard Constraints*

Untuk memastikan bahwa jadwal yang dibuat telah memenuhi *hard constraint* dimana setiap matakuliah harus dijadwalkan dapat dinyatakan dengan persamaan (2)

$$\sum_{i=1}^n \sum_{t=1}^k X_{it} = 1 \quad (2)$$

Dimana  $X_{it}$  adalah matakuliah ke- $i$  yang dijadwalkan pada slot ujian ke- $t$ . Sedangkan untuk memastikan bahwa jadwal yang dibuat telah memenuhi *hardconstraint* kedua dimana tidak ada matakuliah yang mengalami bentrok dinyatakan dengan persamaan (3)

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n C_{ij} \cdot V_{ij} = 0$$

$$V_{ij} \begin{cases} 1 & \text{jika } t_i = t_j \\ 0 & \text{lainnya} \end{cases} \quad (3)$$

Dimana,

$t_i$  = slot ujian dimana matakuliah ke  $i$  dijadwalkan

$C_{ij}$  = jumlah mahasiswa peserta matakuliah  $i$  dan  $j$

Untuk memastikan bahwa jadwal yang dibuat telah memenuhi *hardconstraint* ketiga dimana total mahasiswa peserta ujian



tidak melebihi kapasitas total pada slot ujian ke- $j$  maka dapat digunakan persamaan (4)

$$\sum_{i=1}^n S_i X_{it} \leq \text{kapasitas}$$

(4)

Dimana  $S_i$  adalah jumlah mahasiswa peserta ujian untuk matakuliah ke- $i$

### 5.3.3. Fungsi Tujuan

Fungsi tujuan yang digunakan untuk memastikan bahwa jadwal yang dihasilkan telah optimal atau tidak adalah dengan cara meminimalkan nilai *proximity cost* (P) yang dinyatakan dalam persamaan matematis (5)

$$P = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n C_{ij} W_{|t_j - t_i|}}{M}$$

$$W_{|t_i - t_j|} = \begin{cases} 2^{5-|t_i - t_j|} & \text{jika } 1 \leq |t_i - t_j| \leq 5 \\ 0 & \text{jika } |t_i - t_j| > 5 \end{cases}$$

(5)

Dimana,

$C_{ij}$  = jumlah mahasiswa yang mengikuti matakuliah  $i$  dan matakuliah  $j$ .

$W_{|t_j - t_i|}$  = bobot selisih slot ujian matakuliah  $i$  dan  $j$ .

$M$  = jumlah seluruh mahasiswa.

### 5.3.4. Pembentukan Matriks Jumlah mahasiswa

Untuk mengetahui nilai  $C_{ij}$ , data mahasiswa yang telah didapatkan perlu diubah kedalam sebuah matriks dua dimensi dengan jumlah kolom dan baris sebanyak 32 sesuai dengan jumlah mata kuliah, sehingga didapatkan matriks seperti pada tabel 4.8 dimana setiap sel dari tabel tersebut berisi jumlah mahasiswa yang mengikuti matakuliah  $C_{ij}$ . Selain itu fungsi matriks ini juga untuk mengetahui mata kuliah mana saja yang berpotensi bertabrakan pada saat dijadwalkan

**Tabel 5.8 Potongan Matriks Konflik**

	01	02	03	04	05	06	07	08	09	10
01	0	4	4	4	4	0	0	0	0	0
02	4	0	150	152	151	59	1	8	24	0
03	4	150	0	150	150	57	0	7	24	0
04	4	152	150	0	151	58	1	8	24	0
05	4	151	150	151	0	57	0	7	24	0
06	0	59	57	58	57	0	1	3	2	0
07	0	1	0	1	0	1	0	1	0	0
08	0	8	7	8	7	3	1	0	1	0
09	0	24	24	24	24	2	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0

### 5.3.5. Pembentukan Matriks Selisih Slot

Dengan menggunakan matriks dua dimensi seperti pada tabel 4.8, selisih slot untuk dua matakuliah juga dapat diketahui yang mana nantinya dapat digunakan untuk mencari nilai *proximity cost*. Potongan matrik selisih slot dapat dilihat pada tabel 4.9

**Tabel 5.9 Potongan Matrik Konflik Selisish Slot**

	01	02	03	04	05	06	07	08	09	10
01	0	14	1	6	12	4	13	0	9	3
02	14	0	13	8	2	10	1	14	5	11
03	1	13	0	5	11	3	12	1	8	2
04	6	8	5	0	6	2	7	6	3	3
05	12	2	11	6	0	8	1	12	3	9
06	4	10	3	2	8	0	9	4	5	1
07	13	1	12	7	1	9	0	13	4	10
08	0	14	1	6	12	4	13	0	9	3
09	9	5	8	3	3	5	4	9	0	6
10	3	11	2	3	9	1	10	3	6	0

#### **5.4. Implementasi Algoritma Greedy**

Algoritma *greedy* digunakan untuk menghasilkan jadwal baru berdasarkan data masukan yang telah dipraproses terlebih dahulu. Jadwal yang dihasilkan pada tahapan ini akan menjadi masukan yang akan dioptimasi oleh algoritma *simulated annealing*

#### **5.5. Implementasi Algoritma *Simulated Annealing***

Penerapan algoritma *simulated annealing* ditujukan untuk mengoptimasi jadwal yang telah dihasilkan oleh algoritma *greedy* sebelumnya. Pada tahap ini optimasi dilakukan dengan menukar slot ujian dari dua mata kuliah yang dipilih secara acak atau mengganti slot dari salah satu mata kuliah dengan slot baru yang dipilih secara acak.

*Halaman ini sengaja dikosongkan*

## BAB V IMPLEMENTASI

Pada bab ini berisi tentang proses implementasi algoritma dalam menentukan solusi optimum pada topik tugas akhir ini menggunakan bahasa pemrograman Java pada NetBeans IDE 8.2.

### 6.1. Membaca File Data Input

Terdapat dua macam file yang dijadikan input atau masukkan yaitu file denan format .crs yang berisi data mata kuliah beserta jumlah mahasiswa yang menjadi peserta dari masing-masing mata kuliah dan file format .stu yang berisi data matakuliah yang diambil masing-masing mahasiswa.

```
void bacaCRSSTU() {  
    try {  
  
        FileReader fileReaderCRS = new FileReader(FileCRS);  
        LineNumberReader lineReaderCRS = new LineNumberReader(fileReaderCRS);  
        String lineTextCRS = null;  
  
        while ((lineTextCRS = lineReaderCRS.readLine()) != null) {  
  
            String[] aryCRS = lineTextCRS.split(" ");  
  
            String JumlahMhs = aryCRS[1];  
            int JumlahMhsInt = Integer.parseInt(JumlahMhs);  
            JumlahMhsPerMK.add(JumlahMhsInt);  
  
            TotalMK++;  
        }  
        setTotalMK(TotalMK);  
  
        lineReaderCRS.close();  
    } catch (Exception e) {  
    }
```

**Kode 6.1 Method Baca File crs**

Implementasi pembacaan file .crs dapat dilihat pada potongan kode 5.1 dimana pembacaan ini bertujuan supaya sistem penjadwalan dapat mengetahui jumlah matakuliah dan jumlah mahasiswa pada setiap matakuliah. Untuk file .stu, pembacaan dilakukan supaya sistem mengetahui matakuliah apa saja yang

diambil oleh setiap mahasiswa. Kemudian setiap matakuliah tersebut akan disimpan didalam *array*

```
try {
    FileReader fileReaderSTU = new FileReader(FileSTU);
    LineNumberReader lineReaderSTU = new LineNumberReader(fileReaderSTU);
    String lineTextSTU = null;
    int barisUrut = 0;
    while ((lineTextSTU = lineReaderSTU.readLine()) != null) {
        String[] arySTU = lineTextSTU.split(" ");
        STU[barisUrut] = new int[arySTU.length];
        for (int i = 0; i < arySTU.length; i++) {
            STU[barisUrut][i] = Integer.parseInt(arySTU[i]);
        }
        barisUrut++;
    }
    lineReaderSTU.close();
} catch (Exception e) {
```

**Kode 6.2 Method Baca File stu**

## **6.2. Pembuatan Matriks Konflik**

Pembuatan konflik matriks dilakukan untuk mengetahui jumlah mahasiswa yang mengikuti dua matakuliah. Konflik matriks terdiri dari *array* dua dimensi yang menjadi kolom dan baris. Panjang kolom dan baris dari konflik matriks ini bergantung dari jumlah matakuliah hasil pembacaan dari file .stu

```

int[][] CM() {
    int[][] conflictMatrix = new int[TotalMK][TotalMK];
    for (int i = 0; i < getSTU().length; i++) {
        for (int j = 0; j < getSTU()[i].length - 1; j++) {
            for (int k = j + 1; k < getSTU()[i].length; k++) {
                int arySTi = getSTU()[i][j];
                int arySTj = getSTU()[i][k];
                conflictMatrix[arySTi - 1][arySTj - 1]++;
                conflictMatrix[arySTj - 1][arySTi - 1]++;
            }
        }
    }
    return conflictMatrix;
}

```

Kode 6.3 Method Konflik Matriks

### 6.3. Penerapan Algoritma Greedy

Algoritma *greedy* diterapkan pada pengerjaan tugas akhir ini untuk menghasilkan jadwal ujian awal yang nantinya menjadi input untuk optimasi yang dilakukan oleh algoritma *simulated annealing*. Untuk menentukan suatu ujian dapat dilaksanakan pada suatu slot waktu digunakan dua macam method yaitu okToSlot dan okToRoom. Pada okToSlot matakuliah kedua akan dibandingkan dengan matakuliah sebelumnya untuk melihat apakah terdapat bentrokan

```

boolean okToSlot(int lecture, int slot, int[][] schd) {
    for (int i = 0; i < schd[lecture].length; i++) {
        int ithAdjLecture = schd[lecture][i];
        if (subjectTimeSlot[ithAdjLecture] == slot) {
            return false;
        }
    }
    return true;
}

```

Kode 6.4 Method okToSlot

Berikutnya method okToRoom digunakan untuk membandingkan jumlah mahasiswa peserta suatu matakuliah dengan jumlah kapasitas ruangan yang tersisa yang apabila peserta suatu matakuliah lebih kecil jumlahnya dari kapasitas

pada suatu slot maka akan matakuliah tersebut akan dimasukkan kedalam slot tersebut dan sisa kapasitas pada slot tersebut akan dikurangi dengan jumlah mahasiswa peserta matakuliah tersebut

```
boolean okToRoom(int index, int session, int[] AmountStudentPerSubject) {
    int a = AmountStudentPerSubject[index];
    if (a <= remaincapacity[session - 1]) {
        remaincapacity[session - 1] = remaincapacity[session - 1] - a;
        return true;
    }

    return false;
}
```

**Kode 6.5 Method okToRoom**

Apabila suatu matakuliah telah memenuhi kedua method diatas maka matakuliah tersebut dapat dimasukkan kedalam suatu slot dengan method explore, namun apabila suatu matakuliah tidak memenuhi salah satu atau kedua method maka akan dipindahkan ke slot berikutnya dan akan.

```
boolean explore(int lecture, int slot, int[][] schd, int[] AmountStudentPerSubject) {
    if (lecture >= schd.length) {
        return true;
    }
    int realMK = MK.get(lecture).getIndeks();
    if (okToSlot(realMK, slot, schd) && okToRoom(realMK, slot, AmountStudentPerSubject)) {
        subjectTimeSlot[realMK] = slot;
        for (int i = 1; i <= remaincapacity.length; i++) {
            if (explore(lecture + 1, i, schd, AmountStudentPerSubject)) {
                return true;
            }
        }
    }

    return false;
}
```

**Kode 6.6 Method untuk Menentukan Slot Ujian**

#### 6.4. Penerapan Algoritma Simulated Annealing

Penerapan algoritma *simulated annealing* bertujuan untuk mengoptimasi jadwal yang telah didapat dari algoritma *greedy*. Sebelum dapat menjalankan algoritma ini, beberapa variabel perlu diinisialisasi seperti temperatur awal yang diwakili dengan T, koefisien penurunan Temperatur yang (cooldown), serta



kondisi penghentian algoritma seperti temperatur minimum ( $T_{min}$ ) atau durasi waktu (timer)

```
void SA(int SA, double cooldown, int timer) {
    double T = 1;
    double T_min = 0.00001;
    double alpha = cooldown;
    long start = System.nanoTime();
    long secToNano = 10001 * 1000 * 1000;
```

**Kode 6.7 Inisialisasi Variabel untuk Simulated Annealing**

Jadwal yang telah dihasilkan akan diubah untuk menemukan jadwal yang lebih baik. Untuk mengubah jadwal tersebut dapat dilakukan dengan cara menukar (*swap*) antara dua slot ujian maupun mengubah (*move*) salah satu slot ujian.

```
for (int i = 0; i < SA; i++) {
    if (Math.random() < 0.5) {
        move();
    } else {
        swap();
    }
}
```

**Kode 6.8 Kode untuk Mengubah Slot**

Pada method *move* salah satu slot ujian acak akan diubah slotnya dengan slot acak yang kemudian akan kembali dibandingkan dengan menggunakan method *okToRoom* dan *okToSlot* untuk memastikan bahwa jadwal yang baru telah memenuhi dua method tersebut.

```
void move() {
    int mvRandomLecture = 0;
    this.jadwalBaru = jadwalAwal.clone();
    randomLecture = (int) (Math.random() * subjectTimeSlot.length);
    int randomSlot;
    do {
        randomSlot = (int) ((Math.random() * (remainCapacity.length - 1)) + 1);
    } while (randomSlot == this.jadwalAwal[mvRandomLecture]);
    if (saakToRoom(randomLecture, randomSlot, jadwalBaru)) {
        this.jadwalBaru[randomLecture] = randomSlot;
        int jumlahMhs = AmountStudentPerSubject[randomLecture];
    }
}
```

**Kode 6.9 Method Move**

Pada method swap, dua slot jadwal acak akan ditukar slotnya dan kemudian akan kembali dibandingkan dengan menggunakan method `okToRoom` dan `okToSlot` untuk memastikan bahwa jadwal yang baru telah memenuhi dua method tersebut

```
void swap() {
    int swRandomLecture;
    this.jadwalBaru = jadwalAwal.clone();
    swRandomLecture = (int) (Math.random() * subjectTimeSlot.length);
    int swRandomLecture2 = (int) (Math.random() * subjectTimeSlot.length);
    int swRandomLecture3 = (int) (Math.random() * subjectTimeSlot.length);

    while (swRandomLecture == swRandomLecture2) {
        swRandomLecture2 = (int) (Math.random() * subjectTimeSlot.length);
    }
    if (saokToRoom(swRandomLecture, jadwalBaru[swRandomLecture2], jadwalBaru)
        && saokToRoom(swRandomLecture2, jadwalBaru[swRandomLecture], jadwalBaru)) {
        int temp = jadwalBaru[swRandomLecture];
        jadwalBaru[swRandomLecture] = jadwalBaru[swRandomLecture2];
        jadwalBaru[swRandomLecture2] = temp;
    }
}
```

#### Kode 6.10 Method Swap

Pada algoritma *simulated annealing* jadwal yang telah diubah dengan method *move* atau *swap* akan dibandingkan dengan sebuah *acceptance parameter* acak, apabila jadwal baru memiliki nilai yang lebih tinggi apabila dibandingkan dengan nilai *acceptance parameter* maka jadwal baru tersebut akan diterima.

```

while (T > T_min) {
    //while ((System.nanoTime() - start) < (timer * secToNano)) {
    //while (T > T_min && (System.nanoTime() - start) < (timer * nano)) {
    for (int i = 0; i < SA; i++) {
        if (Math.random() < 0.5) {
            move();
        } else {
            swap();
        }
        double oldSA = saProximity(jadwalAwal.clone());
        double newSA = saProximity(jadwalBaru.clone());
        double ap = Math.exp((oldSA - newSA) / T);
        if (ap > Math.random()) {
            jadwalAwal = jadwalBaru.clone();
        }
    }

    System.out.print(T + " " + saProximity(jadwalAwal) + " ");
    for (int j = 0; j < jadwalAwal.length; j++) {
        System.out.print(jadwalAwal[j] + " ");
    }

    System.out.println("");
    T = T * alpha;
}

```

**Kode 6.11 Simulated Annealing**

Hal ini mengakibatkan jadwal baru yang diterima memiliki kemungkinan untuk lebih buruk apabila dibandingkan dengan jadwal sebelumnya. Apabila jadwal baru telah diterima maka algoritma akan mencari jadwal baru dengan menurunkan temperatur sesuai dengan koefisien penurunannya.

*Halaman ini sengaja dikosongkan*

## **BAB VI HASIL DAN PEMBAHASAN**

Bab ini akan menjelaskan hasil yang didapat dari pengerjaan tugas akhir ini serta pembahasan secara keseluruhan yang didapatkan dari pengerjaan tugas akhir.

### **7.1. Data Uji Coba**

Data yang digunakan adalah data peserta ujian serta mata kuliah tahun ajaran 2016-2017 Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember. Pada data ini diketahui terdapat 567 mahasiswa dan 32 matakuliah.

### **7.2. Lingkungan Uji Coba**

Lingkungan uji coba merupakan kriteria perangkat yang digunakan dalam penerapan algoritma *greedy simulated annealing* pada sistem penjadwalan yang dibangun dalam tugas akhir ini. Lingkungan uji coba terdiri dari perangkat keras dan perangkat lunak. Adapun spesifikasi perangkat keras yang digunakan ditunjukkan pada tabel 6.1

**Tabel 7.1 Spesifikasi Perangkat Keras**

<b>Perangkat Keras</b>	<b>Spesifikasi</b>
Jenis	Notebook
Processor	Intel(R) Core(TM) i5-3230M CPU @ 2,60 GHz
RAM	4 GB
Hardisk	1 TB

Sedangkan untuk spesifikasi perangkat lunak yang digunakan dalam pengujian sistem yang dibuat, ditunjukkan pada tabel 6.2.

**Tabel 7.2 Spesifikasi Perangkat Lunak**

<b>Perangkat Lunak</b>	<b>Spesifikasi</b>
Sistem Operasi	Windows 10
Bahasa Pemrograman	Java
Tools	Netbeans IDE 8.2 Microsoft Excel 2016

### 7.3. Fungsi tujuan dari jadwal manual

Sebelum pembuatan sistem penjadwalan otomatis, dilakukan perhitungan *proximity cost* terhadap jadwal ujian yang telah ada dengan menggunakan *microsoft excel* berdasarkan persamaan (6)

$$P = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N C_{ij} W_{|t_j - t_i|}}{S} \quad (6)$$

Dari hasil perhitungan, diketahui bahwa nilai *proximity cost* untuk jadwal yang dibuat secara manual sebesar 39,569.

Setelah dilakukan perhitungan ulang dengan menggunakan sistem yang dibangun dengan bahasa pemrograman *java* didapati luaran berupa *NOT FEASIBLE* yang menandakan terdapat *hard constraint* yang tidak terpenuhi, dimana pada jadwal UTS manual terdapat dua mata kuliah yang mengalami bentrok jadwal dan dua slot yang mengalami kelebihan kapasitas. Untuk mata kuliah yang mengalami bentrok jadwal terjadi pada mata kuliah dengan kode KS141210 dan KS14108. Jadwal yang bentrok ini menyebabkan 3 orang mahasiswa tidak dapat mengikuti ujian dengan baik. Sedangkan untuk kasus kelebihan kapasitas pada hari Senin slot ke 2 dan hari Jumat slot 2 apabila seluruh ujian dilaksanakan di ruang kelas.

```
NOT FEASIBLE 6 12 Sebanyak 3 mahasiswa
NOT FEASIBLE ROOM CAPACITY EXCEEDED: Timeslot ke 2
NOT FEASIBLE ROOM CAPACITY EXCEEDED: Timeslot ke 14
```

Gambar 7.1 Permasalahan pada Jadwal UTS Manual

```

NOT FEASIBLE 1 10 Sebanyak 1 mahasiswa
NOT FEASIBLE 6 12 Sebanyak 2 mahasiswa
NOT FEASIBLE 6 21 Sebanyak 144 mahasiswa
NOT FEASIBLE 8 12 Sebanyak 1 mahasiswa
NOT FEASIBLE 8 21 Sebanyak 1 mahasiswa
NOT FEASIBLE 10 22 Sebanyak 1 mahasiswa
NOT FEASIBLE 11 13 Sebanyak 141 mahasiswa
NOT FEASIBLE 11 17 Sebanyak 3 mahasiswa
NOT FEASIBLE 12 21 Sebanyak 9 mahasiswa
NOT FEASIBLE 13 17 Sebanyak 7 mahasiswa
NOT FEASIBLE 17 24 Sebanyak 17 mahasiswa
NOT FEASIBLE 22 25 Sebanyak 2 mahasiswa
NOT FEASIBLE 29 30 Sebanyak 8 mahasiswa
NOT FEASIBLE 29 31 Sebanyak 3 mahasiswa
NOT FEASIBLE 30 31 Sebanyak 5 mahasiswa

```

**Gambar 7.2 Permasalahan pada Jadwal UAS Manual**

Pada jadwal UAS manual diketahui bahwa nilai *proximity cost* yaitu sebesar 76,763. Setelah dilakukan perhitungan ulang nilai dengan menggunakan sistem yang dibangun dengan bahasa *java* didapati nilai *proximity cost* sebesar ini disebabkan oleh 15 dari 32 matakuliah yang dijadwalkan saling bertabrakan. Banyaknya jumlah ujian yang saling bertabrakan ini disebabkan jumlah slot ujian yang digunakan hanya terdapat 12 slot ujian dari 15 slot waktu ujian dikarenakan pada minggu ujian terpotong satu hari untuk hari libur nasional

**Tabel 7.3 Mata Kuliah yang Bertabrakan**

Mata Kuliah 1	Mata Kuliah 2	Jumlah Mahasiswa
KS141205	KS141306	1
KS141306	KS141308	2
KS141306	KS141322	144
KS141303	KS141308	1
KS141303	KS141322	1
KS141306	KS141323	1
KS141307	KS141309	141
KS141307	KS141318	3
KS141308	KS141322	9

Mata Kuliah 1	Mata Kuliah 2	Jumlah Mahasiswa
KS141309	KS141318	7
KS141318	KS141402	17
KS141323	KS141403	2

#### 7.4. Hasil Uji Coba Algoritma *Greedy*

Pada bagian ini akan dijelaskan hasil uji coba penggunaan algoritma *greedy* untuk menghasilkan jadwal ujian awal yang kemudian akan dibandingkan dengan hasil jadwal ujian yang dihasilkan secara manual. Jadwal yang dihasilkan oleh algoritma *greedy* nantinya menjadi data masukkan untuk dioptimasi oleh algoritma *simulated annealing*.

##### 7.4.1. Penjadwalan UTS

Algoritma *greedy* digunakan untuk menghasilkan jadwal awal dengan menggunakan data mahasiswa serta matakuliah. Pada uji coba awal pembuatan jadwal, algoritma *greedy* tidak dapat menjadwalkan sepenuhnya matakuliah yang hendak diujikan apabila urutan penjadwalannya berdasarkan urutan matakuliah.

**Tabel 7.4 Algoritma *Greedy* Gagal Menjadwalkan**

Mata Kuliah	Slot Manual	Slot Greedy
KS141204	1	1
KS141205	15	2
KS141206	2	3
KS141207	7	4
KS141208	13	5
KS141209	5	6
KS141210	14	3
KS141301	1	13
KS141303	10	12
KS141304	4	15
KS141306	8	8



<b>Mata Kuliah</b>	<b>Slot Manual</b>	<b>Slot Greedy</b>
KS141307	4	5
KS141308	14	9
KS141309	11	10
KS141310	2	14
KS141316	7	15
KS141317	1	1
KS141318	6	6
KS141319	13	2
KS141320	3	7
KS141321	5	12
KS141322	12	13
KS141323	10	10
KS141325	9	4
KS141402	15	11
KS141403	2	14
KS141404	10	15
KS141405	9	14
KS141407	4	15
KS141409	8	8
KS141411	11	Tidak terjadwal
KS141413	7	Tidak terjadwal

Pada penjadwalan kedua dimana urutan penjadwalan diubah menjadi berdasarkan jumlah kemungkinan terjadinya tabrakan pada setiap matakuliah, algoritma *greedy* berhasil menjadwalkan sepenuhnya seperti yang terlihat pada tabel 6.5

**Tabel 7.5 Perbandingan Hasil Algoritma Greedy dengan Jadwal Manual**

<b>Mata Kuliah</b>	<b>Slot Manual</b>	<b>Slot Greedy</b>
KS141204	1	1
KS141205	15	2

<b>Mata Kuliah</b>	<b>Slot Manual</b>	<b>Slot Greedy</b>
KS141206	2	14
KS141207	7	11
KS141208	13	13
KS141209	5	5
KS141210	14	3
KS141301	1	6
KS141303	10	8
KS141304	4	1
KS141306	8	4
KS141307	4	3
KS141308	14	1
KS141309	11	7
KS141310	2	12
KS141316	7	11
KS141317	1	6
KS141318	6	5
KS141319	13	2
KS141320	3	8
KS141321	5	10
KS141322	12	9
KS141323	10	7
KS141325	9	11
KS141402	15	12
KS141403	2	13
KS141404	10	14
KS141405	9	7
KS141407	4	1
KS141409	8	4
KS141411	11	15

Mata Kuliah	Slot Manual	Slot Greedy
KS141413	7	12

Jadwal yang dihasilkan oleh algoritma *greedy* memiliki nilai *proximity cost* sebesar 60,885. Pada hasil ujian ini sudah tidak ada lagi mahasiswa yang mengalami tabrakan jadwal dan juga tidak ada lagi mata kuliah yang tidak tertampung di kelas, apabila ujian dilaksanakan di seluruh ruang kelas. . Hal ini menunjukkan bahwa hasil jadwal ujian yang menjadi luaran algoritma *greedy* sudah lebih baik apabila dibandingkan dengan jadwal ujian yang dihasilkan secara manual meskipun memiliki nilai *proximity cost* yang lebih buruk apabila dibandingkan dengan jadwal yang dihasilkan secara manual.

#### 7.4.2. Penjadwalan UAS

Pada percobaan menggunakan algoritma *greedy* untuk menghasilkan jadwal UAS menggunakan data mahasiswa dan data matakuliah, percobaan dilakukan dengan 2 skenario, yaitu skenario dengan 12 slot ujian apabila terdapat hari libur seperti yang terjadi pada UAS semester genap dan skenario 15 slot ujian.

Pada saat algoritma *greedy* diterapkan pada skenario 12 slot, algoritma *greedy* gagal untuk menghasilkan jadwal untuk seluruh mata kuliah. Hal ini disebabkan pada saat proses penjadwalan, mata kuliah tersebut tidak dapat memenuhi 2 method penentuan slot untuk memastikan bahwa jadwal nantinya tidak mengalami tabrakan dan masih memenuhi kapasitas total dari seluruh kelas.

Tabel 7.6 Perbandingan Jadwal Hasil Algoritma *Greedy*

Mata Kuliah	Manual	Greedy	
		12 Slot	15 Slot
KS141204	1		2
KS141205	8		11
KS141206	2		13
KS141207	7	11	1

Mata Kuliah	Manual	Greedy	
		12 Slot	15 Slot
KS141208	10		12
KS141209	3	5	4
KS141210	11	2	2
KS141301	1	7	5
KS141303	11	6	6
KS141304	4		1
KS141306	8	8	8
KS141307	4	2	2
KS141308	11	3	3
KS141309	4	10	10
KS141310	2	12	9
KS141316	7	11	11
KS141317	1	4	4
KS141318	4	5	5
KS141319	10	1	1
KS141320	3	6	6
KS141321	5	9	9
KS141322	11	7	7
KS141323	8	10	10
KS141325	9	11	11
KS141402	4		3
KS141403	8		3
KS141404	2		10
KS141405	9		3
KS141407	2	11	12
KS141409			8
KS141411	12	5	13
KS141413		2	12

Pada skenario dengan 15 slot, algoritma *greedy* berhasil menjadwalkan ujian sepenuhnya dengan nilai *proximity cost* sebesar 75,078. Pada skenario ini sudah tidak ada lagi ujian yang mengalami tabrakan dan melebihi kapasitas kelas apabila seluruh ujian dilaksanakan di ruang kelas.

### 7.5. Hasil Uji Coba Algoritma Simulated Annealing

Algoritma *simulated annealing* digunakan untuk mengoptimalkan jadwal ujian hasil luaran dari algoritma *greedy*. Parameter yang digunakan dalam menjalankan algoritma *simulated annealing* ini dapat dilihat pada tabel 6.7

**Tabel 7.7 Parameter Simulated Annealing**

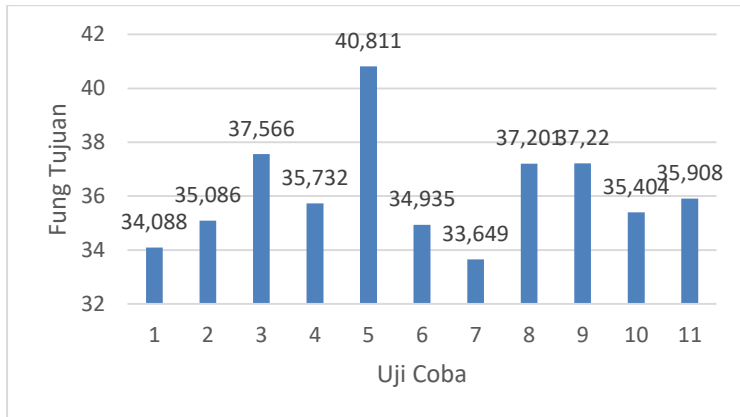
Temperatur awal	1
Koefisien pendinginan	0,99
Temperatur akhir	0,00001
Iterasi	1000
Pemilihan LLH	Random

Temperatur awal menggunakan 1 mengacu pada penelitian terdahulu [8], sedangkan nilai koefisien pendinginan mengacu pada penelitian [10] dimana nilai koefisien pendinginan berada pada nilai  $0,8 \leq a < 1$ , dimana nilai 0,99 dipilih supaya algoritma *simulated annealing* mampu memberikan hasil yang bersifat *global minimum* dengan cara membuat penurunan Temperatur menjadi sangat lambat. Uji coba dilakukan masing-masing sebanyak sebelas kali dikarenakan *simulated annealing* merupakan algoritma yang bersifat stokastik yang melibatkan probabilitas sehingga tidak dapat menghasilkan solusi pasti.

#### 7.5.1. Penjadwalan UTS

Uji coba pembuatan jadwal UTS menggunakan algoritma *simulated annealing* dilakukan sebanyak sebelas kali. Dari sebelas kali uji coba yang dilakukan, algoritma *simulated annealing* mampu menghasilkan jadwal dengan fungsi tujuan (*proximity cost*) hingga 33,649 dengan nilai fungsi tujuan terburuk yaitu sebesar 40,811 dan rata-rata nilai fungsi tujuan

sebesar 36,145. Nilai fungsi tujuan tersebut menunjukkan bahwa jadwal yang dihasilkan oleh algoritma *simulated annealing* berhasil melakukan optimasi dari jadwal ujian yang dihasilkan oleh algoritma *greedy* sebelumnya. Hasil jadwal UTS luaran algoritma *simulated annealing* dapat dilihat pada lampiran 3



**Gambar 7.3 Grafik Fungsi Tujuan dari SA**

**Tabel 7.8 Perbandingan Hasil Jadwal Ujian**

Mata Kuliah	Slot Manual	Slot Greedy	Jadwal SA
KS141204	Senin 1	Senin 1	Rabu 2
KS141205	Jumat 3	Senin 2	Senin 2
KS141206	Senin 2	Jumat 12	Jumat 2
KS141207	Rabu 1	Kamis 2	Selasa 2
KS141208	Jumat 1	Jumat 1	Kamis 2
KS141209	Selasa 2	Selasa 2	Rabu 2
KS141210	Jumat 2	Senin 3	Senin 1
KS141301	Senin 1	Selasa 3	Kamis 3
KS141303	Kamis 1	Rabu 2	Rabu 3
KS141304	Selasa 1	Senin 1	Selasa 1

<b>Mata Kuliah</b>	<b>Slot Manual</b>	<b>Slot Greedy</b>	<b>Jadwal SA</b>
KS141306	Rabu 2	Selasa 1	Jumat 1
KS141307	Selasa 1	Senin 3	Senin 1
KS141308	Jumat 2	Senin 1	Rabu 1
KS141309	Kamis 2	Rabu 1	Kamis 1
KS141310	Senin 2	Kamis 3	Selasa 1
KS141316	Rabu 1	Kamis 2	Selasa 2
KS141317	Senin 1	Selasa 3	Selasa 3
KS141318	Selasa 3	Selasa 2	Senin 3
KS141319	Jumat 1	Senin 2	Jumat 2
KS141320	Senin 3	Rabu 2	Jumat 3
KS141321	Selasa 2	Kamis 1	Rabu 3
KS141322	Kamis 3	Rabu 3	Kamis 3
KS141323	Kamis 1	Rabu 1	Kamis 1
KS141325	Rabu 3	Kamis 2	Kamis 2
KS141402	Jumat 3	Kamis 3	Selasa 2
KS141403	Senin 2	Jumat 1	Selasa 1
KS141404	Kamis 1	Jumat 2	Rabu 2
KS141405	Rabu 3	Rabu 1	Kamis 1
KS141407	Selasa 1	Senin 1	Senin 2
KS141409	Rabu 2	Selasa 1	Jumat 1
KS141411	Kamis 2	Jumat 3	Senin 2
KS141413	Rabu 1	Kamis 3	Selasa 2

Hasil yang didapat melalui optimasi menggunakan algoritma *simulated annealing*, diketahui bahwa terdapat perubahan slot pada jadwal ujian. Pada sebagian ujian perubahan slot ini dapat mengurangi beban mahasiswa dalam mempersiapkan diri sebelum mengikuti ujian. Pada slot manual terdapat sebanyak 154 orang yang harus mengikuti dua ujian yaitu mata kuliah dengan kode KS141317 dan KS141320 pada hari yang sama, setelah dilakukan optimasi beban belajar 154 orang mahasiswa

tersebut dapat berkurang dikarenakan kedua matakuliah tersebut dijadwalkan pada hari yang berbeda yaitu slot Jumat 3 untuk matakuliah KS141320 dan slot Selasa 3 untuk matakuliah KS141317. Beban belajar untuk 159 orang mahasiswa berkurang setelah optimasi dilakukan pada jadwal ujian. Pada jadwal manual 159 orang harus melakukan ujian sebanyak dua kali dalam sehari, yaitu pada matakuliah KS141318 dan KS141321. Setelah dilakukan optimasi matakuliah KS141318 dipindah ke hari senin dan matakuliah KS141321 dipindah ke hari rabu.

**Tabel 7.9 Matakuliah yang Dioptimasi**

<b>Mata Kuliah</b>	<b>Jadwal Manual</b>	<b>Jadwal SA</b>	<b>Jumlah Mahasiswa</b>
KS141205	Jumat 3	Senin 2	151
KS141208	Jumat 1	Kamis 2	
KS141210	Jumat 2	Senin 1	58
KS141319	Jumat 1	Jumat 2	
KS141317	Senin 1	Selasa 3	154
KS141320	Senin 3	Jumat 3	
KS141318	Selasa 3	Senin 3	159
KS141321	Selasa 2	Rabu 3	

Beban belajar untuk 151 orang mahasiswa berkurang setelah optimasi dilakukan pada jadwal ujian. Pada jadwal manual 151 orang harus melakukan ujian sebanyak dua kali dalam sehari pada hari Jumat, yaitu pada matakuliah KS141205 dan KS141208. Setelah dilakukan optimasi matakuliah KS141205 dipindah ke hari senin dan matakuliah KS141208 dipindah ke hari Kamis.

### **7.5.2. Penjadwalan UAS**

Uji coba pembuatan jadwal UAS menggunakan algoritma *simulated annealing* dilakukan sebanyak sebelas kali. Terdapat dua macam jadwal yang dihasilkan, yaitu jadwal untuk 12 slot waktu sesuai dengan kondisi UAS pada semester genap dan 15 slot waktu untuk keadaan dimana minggu ujian tidak terpotong oleh hari libur.



Dari sebelas kali uji coba yang dilakukan untuk 12 slot waktu, algoritma *simulated annealing* tidak mampu menghasilkan jadwal dengan fungsi tujuan (*proximity cost*) yang lebih baik. Nilai fungsi tujuan yang didapat adalah sebesar 184,784 dengan nilai fungsi tujuan terburuk yaitu sebesar 9310,83 dan rata-rata nilai fungsi tujuan sebesar 2730,646. Jadwal yang dihasilkan untuk 12 slot tidak dapat digunakan, karena terdapat empat matakuliah yang mengalami tabrakan jadwal dengan masing-masing satu orang mahasiswa yang mengalami tabrakan.

NOT FEASIBLE 12 30  
NOT FEASIBLE 23 31

Gambar 7.4 Jadwal 12 Slot yang bertabrakan

Tabel 7.10 Jadwal UAS 12 Slot Usulan SA

Mata Kuliah	Slot SA 12 Slot
KS141204	Rabu 3
KS141205	Senin 3
KS141206	Selasa 3
KS141207	Kamis 2
KS141208	Senin 1
KS141209	Selasa 2
KS141210	Senin 2
KS141301	Selasa 1
KS141303	Rabu 1
KS141304	Senin 1
KS141306	Rabu 2
KS141307	Senin 2
KS141308	Selasa 3
KS141309	Kamis 1
KS141310	Rabu 3
KS141316	Kamis 2
KS141317	Selasa 1

Mata Kuliah	Slot SA 12 Slot
KS141318	Selasa 2
KS141319	Senin 1
KS141320	Rabu 2
KS141321	Rabu 3
KS141322	Kamis 3
KS141323	Senin 3
KS141325	Kamis 2
KS141402	Kamis 1
KS141403	Kamis 1
KS141404	Senin 3
KS141405	Kamis 2
KS141407	Selasa 3
KS141409	Rabu 2
KS141411	Selasa 3
KS141413	Kamis 2

Pada uji coba dengan menggunakan 15 slot, algoritma *simulated annealing* berhasil menghasilkan jadwal yang lebih optimal dengan nilai fungsi tujuan terbaik sebesar 34,929 dan rata-rata nilai fungsi tujuan sebesar 37, 566. Hasil jadwal luaran algoritma *simulated annealing* dapat dilihat pada tabel 6.11 dan lampiran 4

**Tabel 7.11 Jadwal UAS 15 Slot Usulan SA**

Mata Kuliah	Slot SA 15 Slot
KS141204	Kamis 1
KS141205	Senin 1
KS141206	Rabu 1
KS141207	Selasa 1
KS141208	Jumat 1
KS141209	Kamis 1
KS141210	Jumat 2

Mata Kuliah	Slot SA 15 Slot
KS141301	Rabu 2
KS141303	Kamis 2
KS141304	Jumat 2
KS141306	Selasa 3
KS141307	Jumat 2
KS141308	Kamis 3
KS141309	Rabu 3
KS141310	Senin 3
KS141316	Selasa 1
KS141317	Selasa 2
KS141318	Rabu 2
KS141319	Senin 1
KS141320	Kamis 2
KS141321	Jumat 1
KS141322	Senin 2
KS141323	Senin 3
KS141325	Selasa 1
KS141402	Rabu 3
KS141403	Rabu 3
KS141404	Senin 3
KS141405	Kamis 2
KS141407	Rabu 1
KS141409	Selasa 3
KS141411	Kamis 1
KS141413	Rabu 1

## 7.6. Hasil dan Pembahasan Uji Coba

Pada bagian ini dilakukan uji coba terhadap algoritma *simulated annealing* dengan cara mengubah nilai koefisien dari parameter

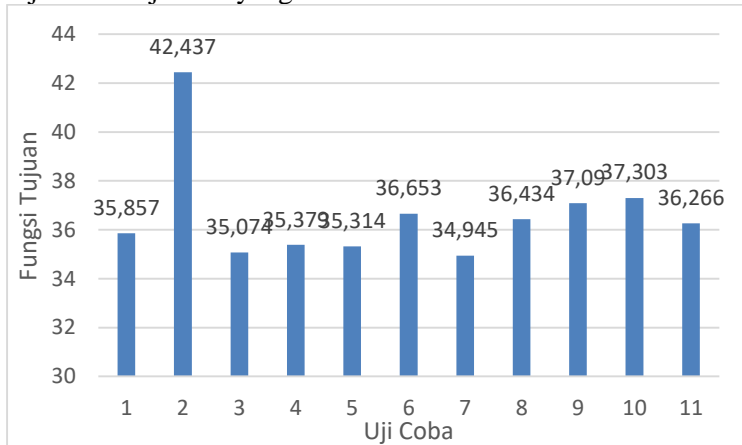
yang digunakan dalam algoritma *simulated annealing*. Tujuan dari dilakukannya uji coba ini adalah untuk melihat pengaruh dari perubahan parameter terhadap jadwal yang dihasilkan oleh algoritma *simulated annealing*. Setiap uji coba dengan perubahan parameter dilakukan sebanyak sebelas kali.

### 7.6.1. Skenario 1 : Perubahan Iterasi Setiap Temperatur

Pada skenario uji coba pertama dilakukan dengan mengubah banyaknya iterasi yang dilakukan pada setiap Temperatur untuk kemudian diteliti apakah jumlah iterasi mempengaruhi fungsi tujuan akhir. Nilai iterasi yang digunakan antara lain: 3000, 5000, 7000 dan 100000. Setiap uji coba dilakukan masing-masing sebanyak sebelas kali. Untuk mendapatkan gambaran rata-rata hasil luaran.

#### 7.6.1.1. Iterasi 3000

Pada uji coba ini nilai iterasi pada setiap Temperatur diubah menjadi 3000 untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan.



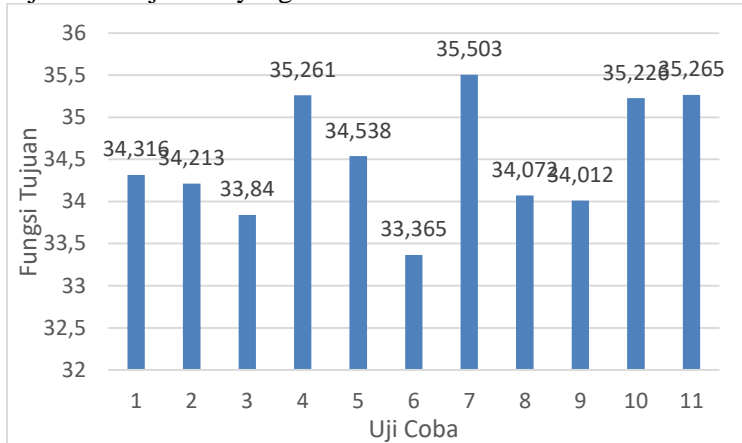
**Gambar 7.5 Fungsi Tujuan dengan Iterasi 3000**

Dari sebelas kali percobaan yang dilakukan dengan mengubah nilai iterasi menjadi 3000 didapatkan nilai fungsi tujuan sebesar 34,945 sebagai fungsi tujuan terbaik dengan nilai rata-rata fungsi tujuan sebesar 36,613. Nilai fungsi tujuan yang didapat

ini tidak lebih baik apabila dibandingkan dengan hasil yang didapat tanpa mengubah parameter iterasi yaitu sebesar 33,649

### 7.6.1.2. Iterasi 5000

Pada uji coba ini nilai iterasi pada setiap Temperatur diubah menjadi 5000 untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan.

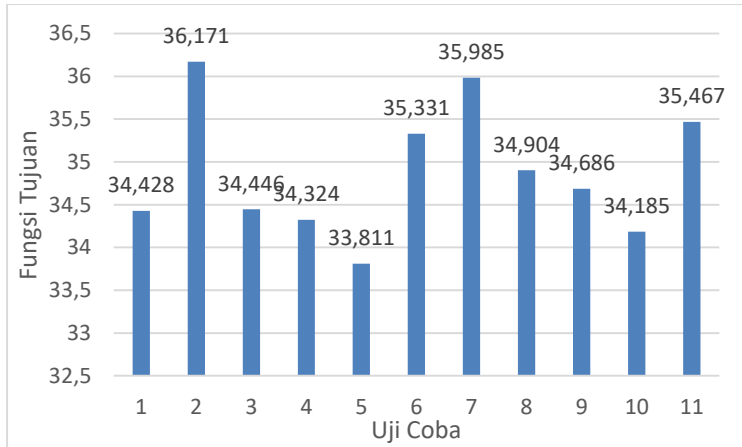


**Gambar 7.6 Fungsi Tujuan dengan Iterasi 5000**

Dari sebelas kali percobaan yang dilakukan dengan mengubah nilai iterasi menjadi 5000 didapatkan nilai fungsi tujuan sebesar 33,365 pada percobaan ke-enam sebagai fungsi tujuan terbaik dengan nilai rata-rata fungsi tujuan sebesar 34,510. Nilai fungsi tujuan yang didapat ini lebih baik apabila dibandingkan dengan hasil yang didapat tanpa mengubah parameter iterasi yaitu sebesar 33,649 ini berarti pada penggunaan iterasi 5000 dapat ditemukan jadwal ujian yang lebih baik lagi

### 7.6.1.3. Iterasi 7000

Pada uji coba ini nilai iterasi pada setiap Temperatur diubah menjadi 7000 untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan.

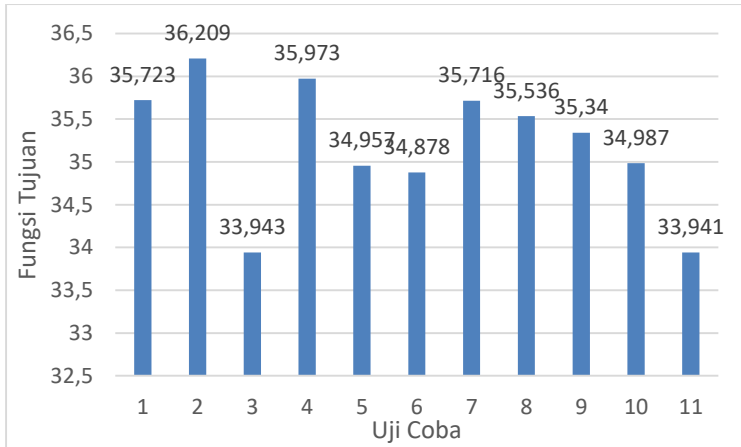


**Gambar 7.7 Fungsi Tujuan dengan Iterasi 7000**

Dari sebelas kali percobaan yang dilakukan dengan mengubah nilai iterasi menjadi 7000 didapatkan nilai fungsi tujuan sebesar 33,811 sebagai fungsi tujuan terbaik dengan nilai rata-rata fungsi tujuan sebesar 34,885. Nilai fungsi tujuan yang didapat ini tidak lebih baik apabila dibandingkan dengan hasil yang didapat tanpa mengubah parameter iterasi yaitu sebesar 33,649

#### **7.6.1.4. Iterasi 100000**

Pada uji coba ini nilai iterasi pada setiap Temperatur diubah menjadi 100000 untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan.



**Gambar 7.8 Fungsi Tujuan dengan Iterasi 100000**

Dari sebelas kali percobaan yang dilakukan dengan mengubah nilai iterasi menjadi 100000 didapatkan nilai fungsi tujuan sebesar 33,941 sebagai fungsi tujuan terbaik dengan nilai rata-rata fungsi tujuan sebesar 35,200. Nilai fungsi tujuan yang didapat ini tidak lebih baik apabila dibandingkan dengan hasil yang didapat tanpa mengubah parameter iterasi yaitu sebesar 33,649

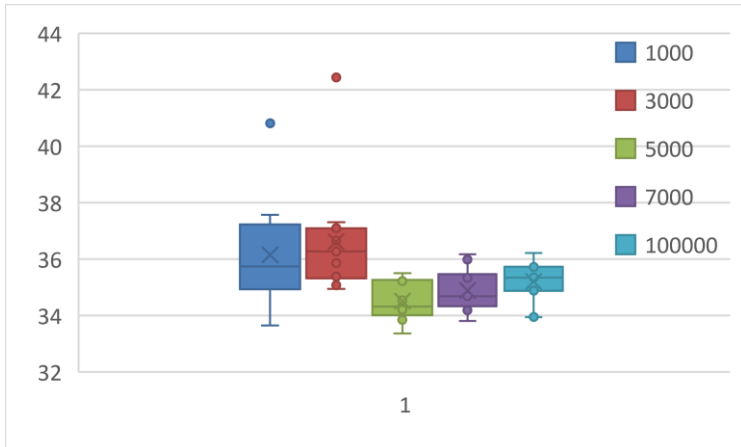
#### 7.6.1.5. Perbandingan Hasil Uji

Hasil uji coba skenario pengubahan iterasi dapat dilihat pada tabel berikut

**Tabel 7.12 Hasil Uji Coba Perubahan Iterasi**

3000	5000	7000	100000
35,857	34,316	34,428	35,723
42,437	34,213	36,171	36,209
35,074	33,84	34,446	33,943
35,379	35,261	34,324	35,973
35,314	34,538	33,811	34,957
36,653	33,365	35,331	34,878
34,945	35,503	35,985	35,716
36,434	34,072	34,904	35,536
37,09	34,012	34,686	35,34

3000	5000	7000	100000
37,303	35,226	34,185	34,987
36,266	35,265	35,467	33,941



**Gambar 7.9 Box-plot Nilai Fungsi Uji Coba**

Dari skenario percobaan yang telah dilakukan, nampak bahwa solusi terbaik didapatkan pada iterasi 5000. Hal ini menunjukkan bahwa nilai iterasi yang semakin besar belum tentu dapat menghasilkan solusi terbaik, namun apabila diperhatikan pada gambar 6.8 nampak bahwa nilai iterasi yang lebih besar dapat menghasilkan solusi yang lebih stabil

**Tabel 7.13 Perbandingan Jadwal dari Perubahan Iterasi**

Jumlah Iterasi	Fungsi Tujuan Terbaik	Usulan Jadwal
3000	34,945	11 2 14 5 8 12 10 1 10 2 11 9 3 14 6 5 1 15 2 13 7 4 6 8 12 9 6 13 9 11 5 12
5000	33,365	8 11 2 14 5 8 1 6 9 1 10 1 4 7 13 2 3 12 14 9 15 6 13 11 8 5 13 9 5 10 2 8



Jumlah Iterasi	Fungsi Tujuan Terbaik	Usulan Jadwal
7000	33,811	4 6 14 10 14 2 12 3 7 5 8 11 14 2 10 4 9 1 7 15 13 12 3 6 10 12 7 11 5 8 6
100000	33,941	1 7 14 10 4 1 11 13 12 2 5 8 2 14 11 4 6 3 9 1 13 15 10 7 4 8 10 1 8 5 12 4

### 7.6.2. Skenario 2 : Pemilihan *Low Level Heuristic*

Pada implementasi algoritma *simulated annealing*, terdapat dua metode pemilihan *low level heuristic* yang digunakan dengan cara dipilih secara acak seperti pada gambar 6.5. Skenario uji coba kedua dilakukan dengan mengubah metode pemilihan *low level heuristic* dengan menentukan bahwa iterasi dilakukan dengan hanya *move*, hanya *swap*, dan random. Skenario ini dilakukan untuk mencari tahu pengaruh pemilihan *low level heuristic* terhadap hasil akhir yang diperoleh dari sistem.

```

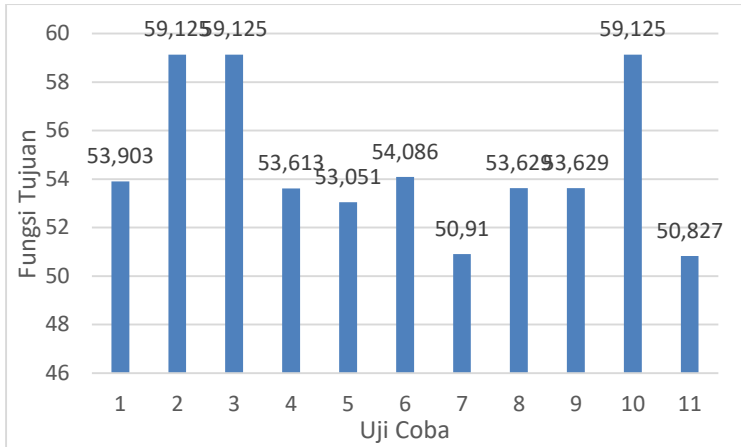
for (int i = 0; i < SA; i++) {
    if (Math.random() < 0.5) {
        move();
    } else {
        swap();
    }
}

```

Gambar 7.10 Pemilihan Low Level Heuristic

#### 7.6.2.1. Pemilihan *Low Level Heuristic* dengan *Move*

Pada uji coba ini pemilihan *low level heuristic* diatur untuk hanya dapat memilih *move* sebagai metodenya. Pada metode *move*, salah satu slot jadwal acak akan diganti dengan slot acak. Hal ini dilakukan untuk melihat apakah terdapat pengaruh antara metode pemilihan *low level heuristic* berupa *move* dengan hasil luaran berupa fungsi tujuan.

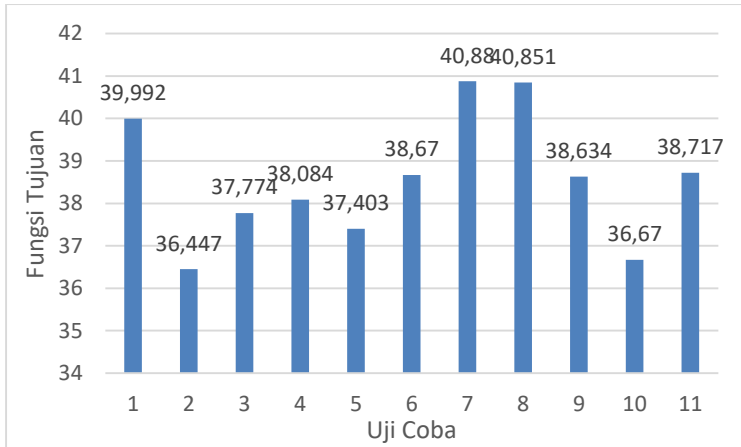


**Gambar 7.11 Fungsi Tujuan *move***

Dari uji coba yang telah dilakukan, didapatkan nilai fungsi tujuan sebesar 50,827 sebagai fungsi tujuan terbaik dengan rata-rata nilai fungsi tujuan sebesar 54,638. Nilai fungsi tujuan yang didapat menandakan bahwa pemilihan *low level heuristic* dengan hanya menggunakan metode *move* kurang mampu melakukan optimasi terhadap jadwal ujian masukkan yang berasal dari algoritma *greedy*.

#### **7.6.2.2. Pemilihan *Low Level Heuristic* dengan *Swap***

Pada uji coba ini pemilihan *low level heuristic* diatur untuk hanya dapat memilih *swap* sebagai metodenya. Pada metode *swap*, dua slot jadwal acak akan ditukar slotnya. Hal ini dilakukan untuk melihat apakah terdapat pengaruh antara metode pemilihan *low level heuristic* berupa *swap* dengan hasil luaran berupa fungsi tujuan.

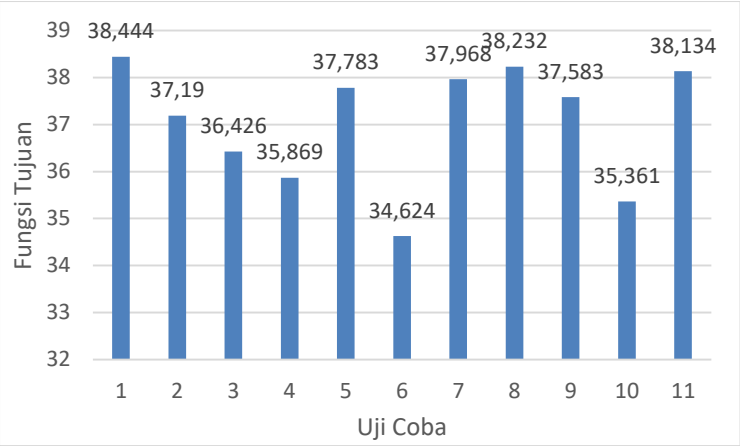


**Gambar 7.12 Fungsi Tujuan *swap***

Dari uji coba yang telah dilakukan, diketahui bahwa pemilihan *low level heuristic* dengan hanya menggunakan metode *swap* sudah mampu menghasilkan jadwal yang lebih optimal bila dibandingkan jadwal masukan dari algoritma *greedy*. Pada uji coba ini didapatkan hasil fungsi tujuan sebesar 36,447 sebagai fungsi tujuan terbaik dengan rata-rata nilai fungsi tujuan sebesar 38,566

### **7.6.2.3. *Low Level Heuristic* Secara Random**

Pada uji coba ini kedua metode pemilihan *low level heuristic*, yaitu *swap* dan *move* akan digunakan secara bersamaan dengan penentuan penggunaan secara acak.



Gambar 7.13 Fungsi Tujuan Acak

Dari uji coba yang telah dilakukan, diketahui bahwa pemilihan *low level heuristic* dengan hanya menggunakan metode *random* mampu menghasilkan jadwal yang lebih optimal bila dibandingkan jadwal masukan dari algoritma *greedy*. Pada uji coba ini didapatkan hasil fungsi tujuan sebesar 34,824 sebagai fungsi tujuan terbaik dengan rata-rata nilai fungsi tujuan sebesar 37,055. Hal ini menunjukkan bahwa dengan menggunakan 2 metode pemilihan *low level heuristic*, algoritma *simulated annealing* dapat menghasilkan jadwal yang lebih optimal apabila dibandingkan dengan hanya menggunakan salah satu metode pemilihan *low level heuristic*.

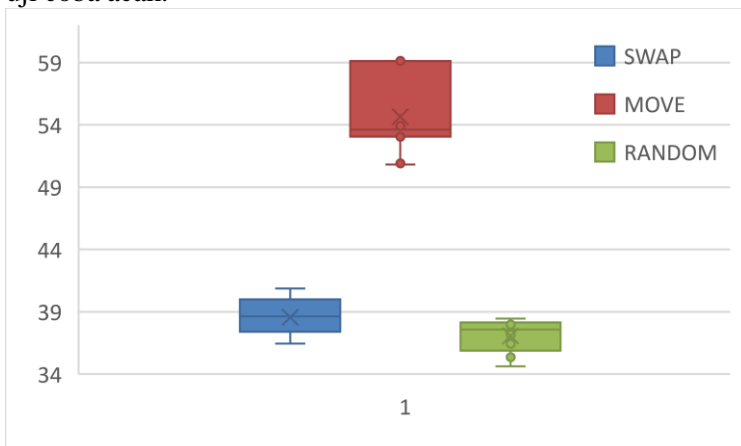
7.6.2.4. Perbandingan Hasil Uji Coba

Tabel 7.14 Perbandingan Hasil Pemilihan LLH

PERCOBAAN	SWAP	MOVE	RANDOM
1	39,992	53,903	38,444
2	36,447	59,125	37,19
3	37,774	59,125	36,426
4	38,084	53,613	35,869
5	37,403	53,051	37,783
6	38,67	54,086	34,624
7	40,88	50,91	37,968

PERCOBAAN	SWAP	MOVE	RANDOM
8	40,851	53,629	38,232
9	38,634	53,629	37,583
10	36,67	59,125	35,361
11	38,717	50,827	38,134
MIN	36,447	50,827	34,624
MAX	40,88	59,125	38,444
AVE	38,55655	54,63845	37,05582

Dari sebelas kali uji coba yang telah dilakukan, nampak bahwa metode *swap* mampu memberikan solusi yang lebih optimal dibandingkan jadwal masukan yang berasal dari algoritma *greedy* meskipun digunakan tanpa metode *move*. Hal berbeda ditunjukkan oleh metode *move*, dimana metode *move* kurang dapat melakukan optimasi apabila digunakan tanpa metode *swap*. Dengan mengkombinasikan metode *swap* dan *move*, algoritma *simulated annealing* mampu menghasilkan fungsi tujuan yang lebih baik seperti yang ditunjukkan pada skenario uji coba acak.



Gambar 7.14 Box-plot Nilai Fungsi Uji Coba

**Tabel 7.15 Perbandingan Jadwal dari Perubahan Metode LLH**

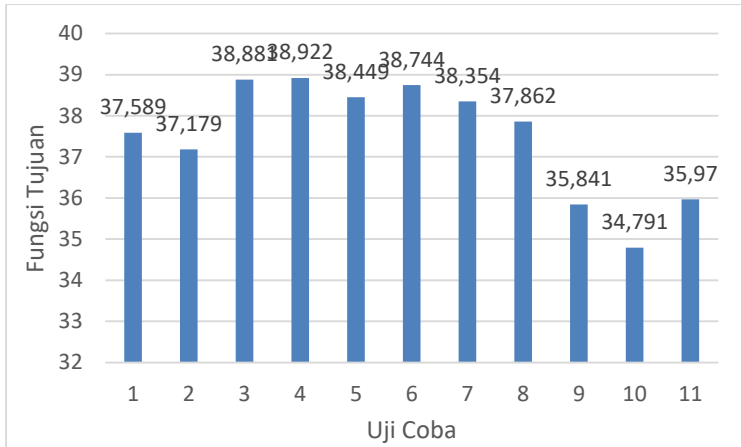
<b>Pemilihan LLH</b>	<b>Fungsi Tujuan Terbaik</b>	<b>Usulan Jadwal</b>
MOVE	50,827	10 2 7 13 3 10 3 9 8 14 4 11 1 7 14 13 6 5 2 8 10 9 14 11 12 13 14 14 1 4 15 12
SWAP	36,447	1 4 8 14 11 2 3 12 6 12 7 3 13 10 1 5 12 9 14 6 15 2 1 4 11 8 1 13 7 5 7 11
RANDOM	34,624	4 1 7 14 10 4 2 12 3 1 5 8 13 11 2 14 9 12 1 4 6 15 7 8 10 14 7 4 14 11 3 10

### **7.6.3. Skenario 3 : Perubahan Koefisien Pendinginan**

Pada skenario uji coba ini dilakukan dengan mengubah nilai koefisien pendinginan Temperatur. Koefisien pendinginan Temperatur yang digunakan pada uji coba ini adalah 0,9, 0,85, dan 0,5. Perubahan koefisien pendinginan akan mempengaruhi kecepatan sistem bekerja, dimana semakin kecil koefisien pendinginan maka semakin cepat pula temperatur pada sistem akan turun yang menyebabkan sistem menghasilkan hasil akhir yang lebih cepat pula. Skenario ini dilakukan untuk mencari tahu pengaruh cepatnya penurunan Temperatur dengan hasil akhir yang diperoleh dari sistem.

#### **7.6.3.1. Koefisien Penurunan Temperatur 0,90**

Pada uji coba ini nilai koefisien penurunan Temperatur diubah menjadi 0,9 untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan.

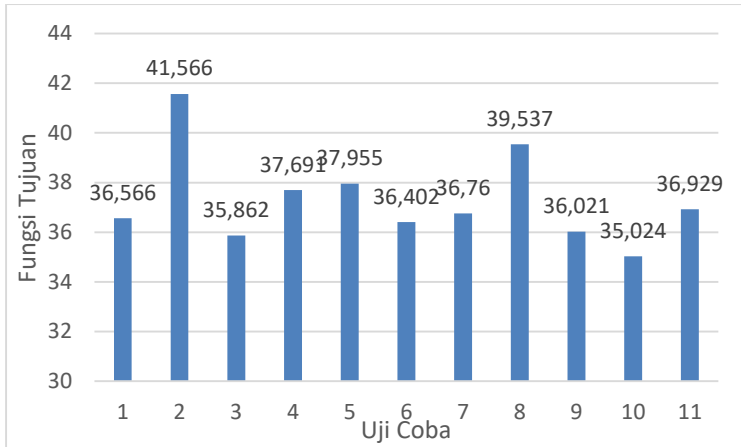


**Gambar 7.15 Fungsi Tujuan Koefisien 0,9**

Dari sebelas kali percobaan yang dilakukan dengan mengubah nilai koefisien penurunan Temperatur menjadi 0,90 didapatkan nilai fungsi tujuan sebesar 34,791 sebagai fungsi tujuan terbaik dengan nilai rata-rata fungsi tujuan sebesar 37,507. Nilai fungsi tujuan yang didapat ini tidak lebih baik apabila dibandingkan dengan hasil yang didapat tanpa mengubah parameter iterasi yaitu sebesar 33,649

#### **7.6.3.2. Koefisien Penurunan Temperatur 0,85**

Pada uji coba ini nilai koefisien penurunan Temperatur diubah menjadi 0,85 untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan



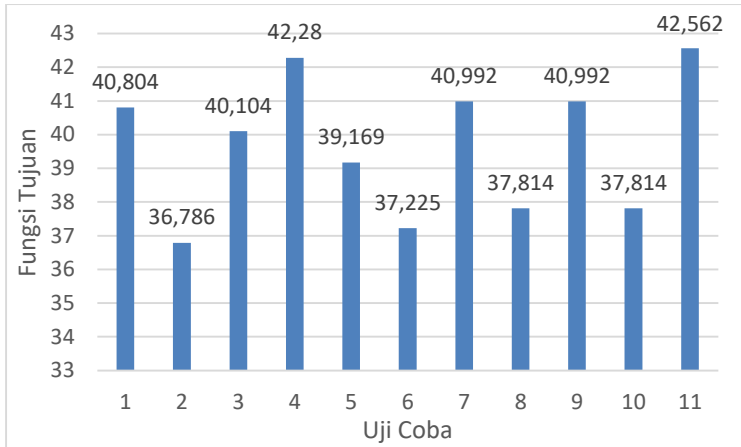
**Gambar 7.16 Fungsi Tujuan Koefisien 0,85**

Dari sebelas kali percobaan yang dilakukan dengan mengubah nilai koefisien penurunan Temperatur sebesar 0,85 didapatkan nilai fungsi tujuan sebesar 35,386 sebagai fungsi tujuan terbaik dengan nilai rata-rata fungsi tujuan sebesar 37,257. Nilai fungsi tujuan yang didapat ini tidak lebih baik apabila dibandingkan dengan hasil yang didapat tanpa mengubah parameter iterasi yaitu sebesar 33,649

#### **7.6.3.3. Koefisien Penurunan Temperatur 0,5**

Pada uji coba ini nilai koefisien penurunan Temperatur diubah menjadi 0,5. Nilai 0,5 dipilih untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan apabila nilai koefisien menggunakan nilai dibawah yang digunakan pada penelitian terdahulu [10].





**Gambar 7.17 Fungsi Tujuan Koefisien 0,5**

Dari sebelas kali percobaan yang dilakukan dengan mengubah nilai iterasi menjadi 7000 didapatkan nilai fungsi tujuan sebesar 33,811 sebagai fungsi tujuan terbaik dengan nilai rata-rata fungsi tujuan sebesar 34,885. Nilai fungsi tujuan yang didapat ini tidak lebih baik apabila dibandingkan dengan hasil yang didapat tanpa mengubah parameter iterasi yaitu sebesar 33,649

#### 7.6.3.4. Perbandingan hasil Uji Coba

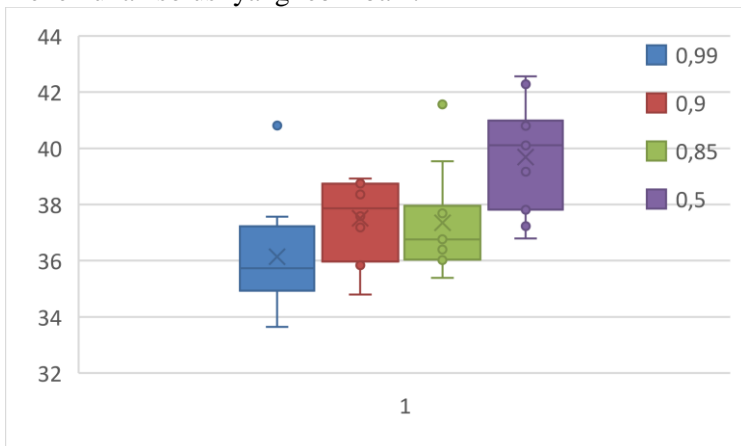
Hasil uji coba perubahan koefisien penurunan Temperatur dapat dilihat pada tabel berikut.

**Tabel 7.16 Perbandingan Hasil Uji Coba Perubahan Koefisien Penurunan Temperatur**

Percobaan	0,99	0,9	0,85	0,5
1	34,088	37,589	36,566	40,804
2	35,086	37,179	41,566	36,786
3	37,566	38,881	35,862	40,104
4	35,732	38,922	37,691	42,28
5	40,811	38,449	37,955	39,169
6	34,935	38,744	36,402	37,225
7	33,649	38,354	36,76	40,992
8	37,201	37,862	39,537	37,814
9	37,22	35,841	36,021	40,992

Percobaan	0,99	0,9	0,85	0,5
10	35,404	34,791	35,024	37,814
11	35,908	35,97	36,929	42,562
MIN	33,649	34,791	35,024	36,786
MAX	40,811	38,922	41,566	42,562
AVE	36,14545	37,50745	37,30118	39,68564

Dari tabel diatas diketahui bahwa perubahan yang dilakukan pada nilai koefisien penurunan Temperatur memiliki pengaruh terhadap nilai rata-rata fungsi tujuan, dimana nilai koefisien pendinginan yang semakin kecil memiliki nilai rata-rata fungsi tujuan yang semakin besar. Hal ini dikarenakan dengan menggunakan nilai koefisien pendinginan yang kecil maka berakibat penurunan Temperatur yang dilakukan oleh sistem menjadi sangat cepat, hal ini mengakibatkan sistem tidak dapat menemukan solusi yang lebih baik.



**Gambar 7.18 Box-plot Uji Coba Perubahan Koefisien Penurunan Temperatur**

**Tabel 7.17 Perbandingan Jadwal hasil Uji Coba**

Koefisien Penurunan Temperatur	Fungsi Tujuan Terbaik	Usulan Jadwal
0,9	34,791	1 4 14 8 11 1 3 13 3 13 5 14 10 7 2 11

Koefisien Penurunan Temperatur	Fungsi Tujuan Terbaik	Usulan Jadwal
		6 9 12 13 15 1 8 4 11 2 8 13 10 5 7 11
0,85	35,386	4 10 2 7 13 5 3 1 4 2 6 3 12 14 9 7 1 5 2 8 11 15 13 10 9 7 13 9 7 14 6 9
0,5	36,786	14 2 8 4 12 10 3 1 13 2 5 3 14 8 11 12 15 1 2 13 9 7 6 4 12 11 6 12 11 5 10 12

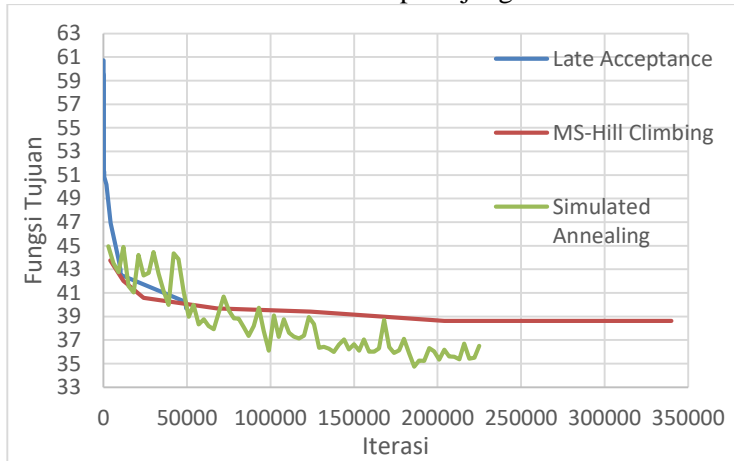
#### 7.6.4. Pembahasan Hasil Skenario Uji Coba

1. Pemilihan jumlah iterasi pada algoritma *simulated annealing* memiliki pengaruh terhadap kinerja sistem. Sistem yang menggunakan jumlah iterasi lebih banyak memiliki nilai fungsi tujuan yang lebih stabil apabila dibandingkan dengan iterasi yang lebih kecil, namun dari uji coba yang telah dilakukan besarnya iterasi tidak mempengaruhi nilai fungsi tujuan akhir.
2. Dari uji coba yang dilakukan, fungsi method *swap* pada pemilihan *low level heuristic* memiliki pengaruh yang signifikan dalam melakukan optimasi jadwal.
3. Koefisien penurunan Temperatur pada algoritma *simulated annealing* memiliki pengaruh pada kinerja sistem optimasi. Dimana koefisien yang kecil mampu menghasilkan jadwal dengan cepat namun kurang optimal apabila dibandingkan dengan nilai koefisien yang lebih besar.

#### 7.7. Perbandingan dengan Algoritma Lain

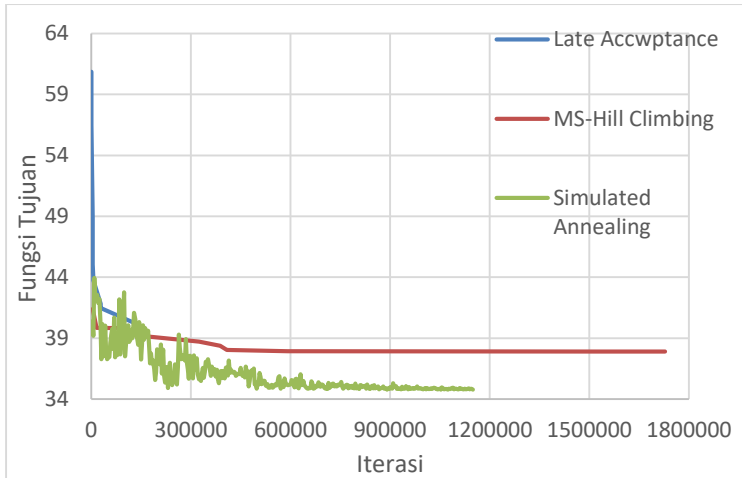
Pada uji coba ini algoritma *simulated annealing* akan dibandingkan dengan dua algoritma optimasi lain, yaitu *late acceptance* dan *multi-stat hill climbing*. Uji coba dilakukan

sebanyak dua kali. Ketiga algoritma dijalankan berdasarkan durasi waktu, yaitu selama 1 menit dan 5 menit. Uji coba ini dilakukan untuk menentukan perbandingan kinerja algoritma dalam menemukan solusi terbaik pada jangka waktu tertentu.



**Gambar 7.19 Grafik Perbandingan Selama 1 Menit**

Pada uji coba yang dilakukan selama satu menit, terlihat bahwa algoritma *simulated annealing* dapat memberikan solusi terbaik dengan nilai fungsi tujuan sebesar 36,499 setelah melakukan 225000 iterasi. Setelah satu menit algoritma *MS-hill climbing* berhasil melakukan 340000 iterasi dengan nilai fungsi tujuan sebesar 38,940 sedangkan algoritma *late acceptance* dalam satu menit hanya dapat melakukan 49515 iterasi dengan nilai fungsi tujuan sebesar 39,677



**Gambar 7.20 Grafik Perbandingan Selama 5 Menit**

Pada uji coba yang dilakukan selama lima menit, terlihat bahwa algoritma *simulated annealing* dapat memberikan solusi terbaik dengan nilai fungsi tujuan sebesar 34,780 setelah melakukan 1149000 iterasi. Setelah lima menit algoritma *MS-hill climbing* berhasil melakukan 17728000 iterasi dengan nilai fungsi tujuan sebesar 37,905 sedangkan algoritma *late acceptance* dalam satu menit hanya dapat melakukan 132385 iterasi dengan nilai fungsi tujuan sebesar 39,921

Dari kedua uji coba yang telah dilakukan dapat diambil kesimpulan bahwa algoritma *simulated annealing* mampu memberikan jadwal yang lebih optimal apabila dibandingkan dengan algoritma *late acceptance* dan *multi stat-hill climbing*.

*Halaman ini sengaja dikosongkan*

## **BAB VII**

### **KESIMPULAN DAN SARAN**

#### **8.1. Kesimpulan**

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Algoritma *greedy simulated annealing* dapat digunakan untuk membuat jadwal ujian yang lebih optimal di Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya dengan menggunakan 15 slot ujian.
2. Dari perbandingan ujicoba yang telah dilakukan diketahui bahwa algoritma *simulated annealing* memiliki kinerja yang lebih baik apabila dibandingkan dengan algoritma *hill climbing* dan *late acceptance*.

#### **8.2. Saran**

Saran yang diberikan berdasarkan hasil pengerjaan tugas akhir untuk penelitian selanjutnya antara lain:

1. Pada pengerjaan tugas akhir ini, setiap satu mata kuliah hanya dijadwalkan untuk satu slot dengan asumsi bahwa setiap mata kuliah melakukan UTS dan UAS. Pada saat pengerjaan diketahui terdapat mata kuliah yang memerlukan lebih dari satu slot dikarenakan mengganti UAS dengan demo atau presentasi FP, sehingga pada penelitian selanjutnya diharapkan sistem penjadwalan mampu menjadwalkan juga demo serta presentasi FP.
2. Pada pemilihan *low level heuristic* bisa ditambahkan dengan metode lain seperti *warming-up approach* atau *step-by-step reduction* [11] untuk mengetahui pengaruh terhadap kinerja algoritma serta hasil akhir dari fungsi tujuan

*Halaman ini sengaja dikosongkan*



## DAFTAR PUSTAKA

- [1] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt Rinehart and Winston, 1976.
- [2] S.-C. Chu dan H. L. Fang, "Genetic algorithms vs. Tabu search in timetable scheduling," dalam *Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*, Adelaide, 1999.
- [3] E. K. Burke, G. Kendall, M. Misir, E. Özcan dan E. Burke, "Applications to timetabling," dalam *Handbook of Graph Theory*, Citeseer, 2004, pp. 445-474.
- [4] R. Bai, E. K. Burke, G. Kendall dan B. McCullum, "A Simulated Annealing Hyper-heuristic for University Course Timetabling Problem," 2006.
- [5] A. Muklason, "Hyper-heuristics and Fairness in Examination Timetabling Problems," The University of Nottingham, Nottingham, 2017.
- [6] E. Samana, B. Prihandono dan E. Noviani, "Aplikasi Simulated Annealing Untuk Menyelesaikan Travelling Salesman Problem," *Buletin Ilmiah Mat. Stat. dan Terapannya*, vol. 03, pp. 25-32, 2015.
- [7] J. Tospornsampan, I. Kita, M. Ishii dan Y. Kitamura, "Split-Pipe Design of Water Distribution Network Using Simulated Annealing," *International Journal of Environmental, Chemical, Ecological, Geological and Geophysical Engineering*, vol. 1, no. 4, pp. 28-38, 2007.
- [8] M. Brusco, H. J. Stolze, M. Hoffman dan D. Steinley, "A Simulated Annealing Heuristic for Maximum Correlation Core/Periphery PARTitioning Binary Network," *Plos One*, vol. 12, no. 5, 2017.
- [9] A. Arram, M. Ayob dan M. Zakree, "Comparative Study of Meta-Heuristic Approaches for Solving Traveling

- Salesman Problems,” *Asian Journal of Applied Sciences*, vol. 7, no. 7, pp. 662-670, 2014.
- [10] E. Soubeiga, “Development and Application of Hyperheuristics to Personnel Scheduling,” University of Nottingham, 2003.
  - [11] K. Chakhlevitch dan P. Cowling, “Choosing the Fittest Subset of Low Level Heuristics in a Hyperheuristic Framework,” University of Bradford, Bradford.
  - [12] N. Bendakir dan E. Aimeur, “Using Association Rules for Course Recommendation,” dalam *AAAI Workshop on Educational Data Mining (Vol. 3)*, Montreal, 2006.
  - [13] Institut Teknologi Sepuluh Nopember, “Histori Status Mahasiswa Per Angkatan,” 02 Mei 2014. [Online]. Available: [http://akademik.its.ac.id/list\\_mhsstatang.php](http://akademik.its.ac.id/list_mhsstatang.php).
  - [14] Institut Teknologi Sepuluh Nopember, “Data Kurikulum Program Studi,” 02 Mei 2014. [Online]. Available: [http://akademik.its.ac.id/data\\_kur.php](http://akademik.its.ac.id/data_kur.php).
  - [15] Institut Teknologi Sepuluh Nopember, “Daftar Prasyarat Mata Kuliah,” 02 Mei 2014. [Online]. Available: [http://akademik.its.ac.id/list\\_prasyarat.php](http://akademik.its.ac.id/list_prasyarat.php).
  - [16] M. W. Carter, G. Laporte dan S. Y. Lee, “Examination timetabling: Algorithmic strategies and applications,” *Journal of the Operational Research Society*, vol. 47, no. 3, pp. 373-383, 1996.
  - [17] A. Muklason, “Solver Penjadwal Ujian Otomatis Dengan Algoritma Maximal Clique dan Hyper-heuristics,” dalam *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI) 9, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau*, Pekanbaru, 2017.
  - [18] A. Schaerf, “A survey of automated timetabling,” *Artificial Intelligence Review*, vol. 12, no. 2, pp. 87-127, April 1999.
  - [19] E. K. Burke, G. Kendall, M. Misir dan E. Özcan, “Monte Carlo hyper-heuristics for examination timetabling,”

*Annals of Operations Research*, vol. 196, no. 1, pp. 73-90, 2012.

- [20] E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic dan R. Qu, "Hybrid variable neighbourhood approaches to university exam timetabling," *European Journal of Operational Research*, vol. 206, no. 1, pp. 46-53, 2010.
- [21] T. B. Cooper dan J. H. Kingston, "The complexity of timetable construction problems. In Edmund Burke and Peter Ross, editors, Practice and Theory of Automated Timetabling," *Springer Berlin Heidelberg*, vol. 1153 of Lecture Notes in Computer Science, pp. 281-295, 1996.
- [22] University of Toronto, "University of Toronto Benchmark Data," 1996. [Online]. Available: <ftp://ftp.mie.utoronto.ca/pub/carter/testprob>. [Diakses 26 September 2017].
- [23] R. Munir, "Algoritma Greedy," Program Studi Informatika, Sekolah Teknik Elektro dan Informatika ITB, Bandung, 2004.
- [24] R. Hartanto, "Algoritma Greedy dalam Strategi Permainan Centipede," Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika ITB, Bandung, 2017.
- [25] E. Özcan, Y. Bykov, M. Birben dan E. K. Burke, "Examination Timetabling Using Late Acceptance Hyper-heuristics," dalam *IEEE Congress on Evolutionary Computation CEC '09*, 2009.
- [26] E. K. Burke, G. Kendall, J. Newall, E. Hart, P. Ross dan S. Schulenburg, "Hyper-Heuristics: An emerging direction in modern search. In Fred Glover and Gary A. Kochenberger, editors, Handbook of Metaheuristics," *International Series in Operations Research & Management. Springer US.*, vol. 57, pp. 457-474, 2003.
- [27] S. Kirkpatrick, C. D. Gelatt dan M. P. Vecchi, "Optimization by Simulated Annealing," *Science, New Series*, vol. 220, no. 4598, pp. 671-679, 1983.

- [28] A. Muklason, A. J. Parker, E. Ozcan, B. McCollum dan P. McMullan, "Fairness in examination timetabling: Student preferences and extended formulations," *Applied Soft Computing*, pp. 302-317, 2015.

## RIWAYAT PENULIS



Dian Kusumawardani dilahirkan di Surabaya pada tanggal 2 Mei 1992. Menempuh pendidikan formal yakni SD Negeri Keputran 6 Surabaya, dan SMP Negeri 3 Surabaya. Jenjang pendidikan berikutnya penulis memilih untuk menempuh di SMA Negeri 6 Surabaya dan lulus pada tahun 2010. Pada tahun yang sama, penulis melanjutkan pendidikan ke bangku perguruan tinggi di Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember Surabaya

setelah dinyatakan lolos melalui jalur SNMPTN dengan NRP 5210100127.

Di Jurusan Sistem Informasi ini penulis mengambil bidang minat Rekayasa Data dan Intelijensi Bisnis dengan topik tugas akhir adalah penjadwalan ujian.

Penulis dapat dihubungi melalui email [kusuma.dian@live.com](mailto:kusuma.dian@live.com) apabila diperlukan untuk kepentingan penelitian.

*Halaman ini sengaja dikosongkan*

## LAMPIRAN 1

Lampiran ini merupakan hasil pemetaan dari data mahasiswa terhadap matakuliah setelah dilakukan praproses. Data ini disimpan pada file .stu

**Tabel 11.1 Pemetaan Mahasiswa Terhadap Matakuliah**

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0001	0023
0002	0010 0014
0003	0032
0004	0013 0022
0005	0007 0011 0013 0014 0025 0026 0027
0006	0031
0007	0021 0022 0025
0008	0013 0017 0018 0023 0024
0009	0011
0010	0031
0011	0007 0011 0018 0020 0022 0024 0025
0012	0007 0017 0018 0023
0013	0023 0029
0014	0007 0017 0019 0021 0022 0024 0029
0015	0025
0016	0007 0027
0017	0031
0018	0007 0011 0013 0014 0024
0019	0009 0014 0015 0018 0019
0020	0009 0011 0012 0017 0019 0022 0032
0021	0006 0013 0030
0022	0017
0023	0022 0027

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0024	0013 0018 0019 0023 0032
0025	0004 0011 0014 0017 0020
0026	0013 0022 0031
0027	0021 0022 0030
0028	0013 0022 0024
0029	0032
0030	0017
0031	0031
0032	0025 0031
0033	0017 0031
0034	0027
0035	0018 0021 0022 0030 0031
0036	0032
0037	0031
0038	0013 0019 0020 0021 0032
0039	0031
0040	0017 0028
0041	0032
0042	0031
0043	0031
0044	0032
0045	0027 0031
0046	0007 0018 0021 0030 0031
0047	0007 0017 0021 0022 0031 0032
0048	0032
0049	0017 0020 0021 0022
0050	0032
0051	0020
0052	0031



<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0053	0030 0031
0054	0012
0055	0011 0012 0016 0017 0021 0023
0056	0025
0057	0002 0004 0006 0007 0008 0014
0058	0006 0009 0013
0059	0032
0060	0007 0017 0028
0061	0030 0031
0062	0024 0031
0063	0023 0025 0030
0064	0027 0031
0065	0030 0031
0066	0013 0017 0018 0019 0021 0030
0067	0011 0013 0014 0019 0020 0024
0068	0018 0027
0069	0027
0070	0030
0071	0026 0027 0030 0031 0032
0072	0017 0022 0023 0026 0032
0073	0020 0022
0074	0017 0021 0030 0031
0075	0031 0032
0076	0026 0030 0031 0032
0077	0018 0031
0078	0007 0017 0018 0021 0024 0032
0079	0007 0017 0018 0021 0022 0028 0030
0080	0007 0017 0021 0022 0031 0032
0081	0031

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0082	0021
0083	0025
0084	0021
0085	0017 0019 0021 0022 0023 0031
0086	0030
0087	0026
0088	0007 0031 0032
0089	0030 0031
0090	0021 0031
0091	0031
0092	0017 0018 0022
0093	0017
0094	0007 0017 0018 0019 0020 0021 0022 0030
0095	0007 0017 0018 0019 0020 0021 0022
0096	0013 0018 0020 0021 0022 0024
0097	0007 0017 0018 0019 0022 0023
0098	0007 0017 0018 0019 0020 0021 0022
0099	0007 0017 0018 0020 0021 0022 0030
0100	0007 0017 0018 0019 0020 0021 0022
0101	0007 0017 0018 0020 0021 0022 0024 0026
0102	0007 0017 0018 0020 0021 0022 0026 0030
0103	0007 0017 0018 0020 0021 0022 0025 0027
0104	0007 0017 0018 0020 0021 0022 0026
0105	0007 0017 0019 0020 0021 0022 0025 0029
0106	0007 0017 0018 0019 0020 0021 0022 0024

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0107	0007 0017 0018 0019 0020 0021 0022
0108	0007 0017 0018 0019 0020 0021 0022
0109	0007 0017 0018 0019 0020 0021 0022 0027
0110	0007 0017 0018 0019 0020 0021 0022
0111	0007 0017 0018 0020 0021 0022 0027
0112	0017 0018 0021 0022 0027 0030
0113	0007 0017 0018 0019 0020 0021 0022
0114	0007 0017 0018 0020 0021 0022 0030
0115	0007 0017 0018 0020 0021 0022 0023 0025
0116	0007 0013 0017 0018 0020 0021 0022
0117	0007 0017 0018 0019 0020 0021 0022
0118	0007 0017 0020 0021 0022 0030 0032
0119	0017 0018 0019 0020 0021 0022 0027
0120	0007 0017 0018 0020 0021 0022 0032
0121	0007 0017 0018 0019 0020 0021 0022 0024
0122	0007 0017 0018 0019 0020 0021 0022 0030
0123	0007 0017 0018 0020 0021 0022 0032
0124	0007 0017 0018 0019 0020 0021 0022
0125	0007 0017 0018 0020 0021 0022 0023 0026
0126	0007 0017 0018 0020 0021 0022 0024 0026
0127	0007 0017 0018 0020 0021 0022 0023 0032
0128	0007 0017 0018 0019 0020 0021 0022 0029
0129	0007 0017 0020 0021 0022 0026
0130	0013 0014 0017 0018 0020

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0131	0007 0017 0018 0019 0020 0021 0022
0132	0007 0017 0018 0020 0021 0022 0025 0029
0133	0008 0011 0012 0014 0016
0134	0007 0017 0018 0019 0020 0021 0022 0024
0135	0007 0017 0018 0019 0020 0021 0022 0023
0136	0007 0017 0018 0020 0021 0022 0024
0137	0011 0012 0013 0014 0020 0022
0138	0007 0017 0018 0020 0021 0022 0032
0139	0007 0017 0018 0020 0021 0022 0027 0030
0140	0007 0017 0018 0020 0021 0022 0026 0027
0141	0007 0017 0018 0020 0021 0022 0023
0142	0007 0017 0018 0020 0021 0022 0024 0029
0143	0007 0017 0018 0019 0020 0021 0022 0026
0144	0007 0017 0018 0019 0020 0021 0022
0145	0007 0017 0018 0020 0021 0022 0026 0027
0146	0007 0017 0018 0020 0021 0022 0030
0147	0007 0017 0018 0019 0020 0021 0022 0023
0148	0007 0017 0018 0020 0021 0022 0023 0032
0149	0007 0017 0018 0020 0021 0022 0027 0030
0150	0007 0017 0018 0020 0021 0022 0027 0030
0151	0007 0017 0018 0020 0021 0022 0027 0030

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0152	0007 0017 0018 0020 0021 0022 0032
0153	0007 0017 0018 0021 0022 0030 0032
0154	0007 0017 0018 0020 0021 0022 0023 0029
0155	0007 0017 0018 0020 0021 0022 0030
0156	0007 0017 0018 0020 0021 0022 0025
0157	0007 0017 0018 0020 0021 0022 0023 0032
0158	0007 0017 0018 0020 0021 0022 0032
0159	0007 0014 0016 0017 0020 0022
0160	0007 0017 0018 0020 0021 0022 0025
0161	0007 0017 0018 0020 0021 0022 0023 0031
0162	0007 0017 0018 0020 0021 0022 0023 0029
0163	0007 0017 0018 0019 0020 0021 0022 0023
0164	0007 0017 0018 0019 0020 0021 0022 0029
0165	0007 0017 0018 0020 0021 0022 0032
0166	0007 0017 0018 0019 0020 0021 0022 0026
0167	0007 0017 0018 0020 0021 0022 0032
0168	0007 0017 0018 0020 0021 0022 0026 0027
0169	0007 0017 0018 0020 0021 0022 0029
0170	0007 0017 0018 0021 0022 0025 0030
0171	0007 0017 0018 0020 0021 0022 0024 0027
0172	0007 0017 0018 0020 0021 0022 0027 0030
0173	0007 0017 0018 0019 0020 0021 0022
0174	0007 0017 0018 0019 0020 0021 0022

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0175	0011 0013 0014 0017 0020 0024
0176	0007 0020 0021
0177	0017 0018 0020 0021 0022 0029
0178	0017 0018 0019 0020 0021 0022
0179	0017 0018 0019 0020 0021 0022
0180	0017 0018 0019 0020 0021 0022
0181	0013 0017 0019 0020 0021 0022
0182	0007 0017 0018 0019 0021 0022 0026
0183	0017 0018 0020 0021 0022 0025 0029
0184	0017 0019 0020 0021 0022 0030 0031
0185	0007 0014 0017 0018 0019 0020 0022
0186	0007 0017 0018 0020 0021 0022 0032
0187	0007 0017 0018 0020 0021 0022 0023 0024
0188	0007 0017 0018 0020 0021 0022 0023 0030
0189	0007 0017 0020 0021 0022 0026 0027
0190	0007 0017 0018 0020 0021 0022 0027
0191	0007 0017 0018 0020 0021 0022 0024
0192	0007 0017 0018 0020 0021 0022 0026
0193	0017 0018 0020 0021 0022 0023
0194	0011 0012 0014 0015 0019 0020
0195	0017 0018 0019 0021 0022 0028 0029
0196	0007 0017 0018 0020 0021 0022 0024 0030
0197	0007 0017 0018 0019 0020 0021 0022
0198	0012 0014 0017 0018 0019 0020 0022
0199	0007 0017 0018 0020 0021 0022 0027
0200	0007 0017 0018 0020 0021 0022 0030 0032
0201	0007 0017 0018 0019 0020 0021 0022

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0202	0007 0017 0020 0021 0022 0023 0025 0029
0203	0007 0017 0018 0020 0021 0022 0024 0027
0204	0007 0017 0018 0020 0021 0022 0027 0030
0205	0007 0017 0018 0019 0020 0021 0022
0206	0007 0017 0018 0019 0020 0021 0022 0029
0207	0007 0017 0018 0020 0021 0022 0026 0027
0208	0007 0017 0018 0019 0020 0021 0022
0209	0007 0017 0018 0020 0021 0022 0024 0029
0210	0007 0017 0018 0020 0021 0022 0026 0027
0211	0007 0017 0018 0020 0021 0022 0026 0027
0212	0011 0013 0019 0020 0021 0022
0213	0007 0017 0018 0020 0021 0022 0025 0027
0214	0007 0017 0018 0019 0020 0021 0022
0215	0007 0017 0018 0019 0020 0021 0022 0024
0216	0007 0017 0018 0019 0020 0021 0022 0027
0217	0011 0017 0018 0019 0020 0021 0022
0218	0007 0017 0018 0020 0021 0022 0025
0219	0007 0017 0018 0019 0020 0021 0022
0220	0007 0017 0018 0019 0020 0021 0022
0221	0007 0017 0018 0019 0020 0021 0022
0222	0007 0017 0018 0019 0020 0021 0022
0223	0007 0017 0018 0019 0020 0021 0022

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0224	0007 0017 0018 0020 0021 0022 0026 0027
0225	0007 0017 0018 0020 0021 0022 0032
0226	0017 0018 0019 0020 0021 0022 0025
0227	0007 0017 0018 0020 0021 0022 0025 0030
0228	0007 0017 0018 0020 0021 0022 0027
0229	0017 0018 0019 0020 0021 0022 0030
0230	0007 0017 0018 0019 0020 0021 0022
0231	0007 0017 0018 0019 0020 0021 0022
0232	0011 0013 0014 0018 0020 0022
0233	0007 0017 0018 0021 0022 0030 0031
0234	0007 0017 0018 0019 0020 0021 0022
0235	0011 0014 0018 0020 0022
0236	0007 0017 0018 0020 0021 0022 0030
0237	0007 0017 0018 0020 0021 0022 0026 0030
0238	0007 0016 0017 0018 0019 0021 0022
0239	0007 0017 0018 0020 0021 0022 0027
0240	0007 0017 0018 0019 0020 0021 0022
0241	0007 0017 0018 0019 0021 0022 0032
0242	0007 0017 0018 0021 0022 0024 0025 0027
0243	0007 0017 0018 0021 0022 0029 0030
0244	0017 0018 0019 0021 0022 0024 0029
0245	0007 0017 0018 0020 0021 0022 0032
0246	0007 0017 0018 0021 0022 0026 0030
0247	0007 0017 0018 0021 0022 0026 0032
0248	0007 0017 0018 0020 0021 0022 0030
0249	0007 0017 0018 0021 0022 0024 0025 0027



<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0250	0007 0017 0018 0020 0021 0022 0024 0025
0251	0007 0017 0019 0020 0021 0022 0030
0252	0007 0017 0019 0020 0021 0022 0030
0253	0007 0017 0018 0019 0020 0021 0022
0254	0017 0018 0019 0020 0021 0022 0031
0255	0007 0017 0018 0020 0021 0022 0027
0256	0017 0018 0020 0021 0022 0027 0030
0257	0011 0018 0019 0020 0022
0258	0007 0017 0018 0019 0020 0021 0022
0259	0017 0018 0019 0020 0021 0022 0032
0260	0017 0018 0020 0021 0022 0030 0031
0261	0017 0018 0019 0020 0021 0022 0032
0262	0012 0017 0018 0019 0020 0021 0022
0263	0017 0018 0019 0020 0021 0022 0026
0264	0007 0017 0018 0019 0020 0021 0022 0025
0265	0007 0017 0018 0021 0022 0026 0030
0266	0007 0017 0018 0020 0021 0022 0023 0032
0267	0007 0017 0018 0019 0020 0021 0022 0025
0268	0011 0013 0019 0020 0021 0022
0269	0007 0017 0018 0019 0020 0021 0022 0023
0270	0008 0011 0013 0015 0019
0271	0007 0017 0018 0019 0020 0021 0022 0024
0272	0017 0018 0019 0020 0021 0022 0024 0025
0273	0011 0012 0013 0014 0015 0019
0274	0011 0012 0013 0014 0015 0016

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0275	0004 0011 0012 0013 0014 0015
0276	0011 0012 0013 0014 0015
0277	0011 0012 0013 0014 0015 0019
0278	0011 0012 0013 0014 0015 0019
0279	0011 0012 0013 0014 0015 0019
0280	0011 0012 0013 0014 0015 0019
0281	0011 0012 0013 0014 0015
0282	0011 0012 0013 0014 0015 0019
0283	0008 0011 0012 0013 0014 0015
0284	0011 0012 0013 0014 0015
0285	0011 0012 0013 0014 0015 0019
0286	0011 0012 0013 0014 0015 0019
0287	0011 0012 0013 0014 0015
0288	0011 0012 0013 0014 0015 0019
0289	0011 0012 0013 0014 0015
0290	0002 0012 0014 0016 0018
0291	0011 0012 0013 0014 0015 0019
0292	0011 0012 0013 0014 0015
0293	0011 0012 0013 0014 0015
0294	0011 0012 0013 0014 0015
0295	0011 0012 0013 0014 0015
0296	0011 0012 0013 0014 0015 0019
0297	0011 0012 0013 0014 0015 0016
0298	0011 0012 0013 0014 0015 0016
0299	0011 0012 0013 0014 0015
0300	0011 0012 0013 0014 0015
0301	0011 0012 0013 0014 0015 0016
0302	0011 0012 0013 0014 0015 0019
0303	0011 0012 0013 0014 0015

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0304	0011 0012 0013 0014 0015 0019
0305	0011 0012 0013 0014 0015
0306	0011 0012 0013 0014 0015
0307	0011 0012 0013 0014 0015
0308	0011 0012 0013 0014 0015
0309	0011 0012 0013 0014 0015 0019
0310	0011 0012 0013 0014 0015
0311	0011 0012 0013 0014 0015 0016
0312	0011 0012 0013 0014 0015
0313	0011 0012 0013 0014 0015
0314	0011 0012 0013 0014 0015
0315	0011 0012 0013 0014 0015
0316	0011 0012 0013 0014 0015 0019
0317	0011 0012 0013 0014 0015
0318	0011 0012 0013 0014 0015
0319	0011 0012 0013 0014 0015 0020
0320	0011 0012 0013 0014 0015 0019
0321	0011 0012 0013 0014 0015
0322	0011 0012 0013 0014 0015 0019
0323	0011 0012 0013 0014 0015
0324	0011 0012 0013 0014 0015 0016
0325	0011 0012 0013 0014 0015 0019
0326	0011 0012 0013 0014 0015
0327	0011 0012 0013 0014 0015 0019
0328	0011 0012 0013 0014 0015
0329	0008 0011 0013 0014 0015
0330	0011 0012 0013 0014 0015
0331	0011 0012 0013 0014 0015
0332	0011 0012 0013 0014 0015

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0333	0011 0012 0013 0014 0015
0334	0011 0012 0013 0014 0015
0335	0011 0012 0013 0014 0015
0336	0011 0012 0013 0014 0015 0019
0337	0011 0012 0013 0014 0015
0338	0011 0012 0013 0014 0015 0019
0339	0011 0012 0013 0014 0015
0340	0011 0012 0013 0014 0015 0019
0341	0011 0012 0013 0014 0015
0342	0011 0012 0013 0014 0015
0343	0011 0012 0013 0014 0015 0019
0344	0011 0012 0013 0014 0015 0019
0345	0011 0012 0013 0014 0015 0019
0346	0011 0012 0013 0014 0015 0016
0347	0011 0012 0013 0014 0015
0348	0011 0012 0013 0014 0015
0349	0011 0012 0013 0014 0015
0350	0011 0012 0013 0014 0015
0351	0011 0012 0013 0014 0015 0019
0352	0011 0012 0013 0014 0015 0019
0353	0011 0012 0013 0014 0015 0019
0354	0011 0012 0013 0014 0015 0019
0355	0011 0012 0013 0014 0015 0016
0356	0011 0012 0013 0014 0015 0016
0357	0011 0012 0013 0014 0015
0358	0011 0012 0013 0014 0015
0359	0011 0012 0013 0014 0015 0019
0360	0011 0012 0013 0014 0015
0361	0011 0012 0013 0014 0015 0019

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0362	0011 0012 0013 0014 0015
0363	0011 0012 0013 0014 0015 0019
0364	0006 0008 0011 0012 0015 0019
0365	0011 0012 0013 0014 0015
0366	0011 0012 0013 0014 0015
0367	0008 0011 0012 0013 0014 0015
0368	0011 0012 0013 0014 0015 0019
0369	0006 0011 0012 0014 0015
0370	0011 0012 0013 0014 0015
0371	0011 0012 0013 0014 0015
0372	0011 0012 0013 0014 0015
0373	0011 0012 0013 0014 0015 0019
0374	0011 0012 0013 0014 0015 0016
0375	0011 0012 0013 0014 0015 0019
0376	0011 0012 0013 0014 0015 0019
0377	0008 0011 0012 0013 0015
0378	0011 0012 0013 0014 0015
0379	0011 0012 0013 0014 0015 0019
0380	0011 0012 0013 0014 0015 0016
0381	0011 0012 0013 0014 0015
0382	0002 0006 0012 0014
0383	0011 0012 0013 0014 0015
0384	0011 0012 0013 0014 0015
0385	0002 0004 0005 0014
0386	0006 0008 0009 0012 0016
0387	0011 0012 0013 0014 0015
0388	0011 0012 0013 0014 0015 0019
0389	0011 0012 0013 0014 0015 0019 0020
0390	0011 0012 0013 0014 0015 0020

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0391	0011 0012 0013 0014 0015 0019
0392	0011 0012 0013 0014 0015 0019
0393	0011 0012 0013 0014 0015 0020
0394	0002 0011 0012 0013 0014 0015
0395	0011 0012 0013 0014 0015
0396	0011 0012 0013 0014 0015
0397	0011 0012 0013 0014 0015 0016
0398	0011 0012 0013 0014 0015 0019
0399	0011 0012 0013 0014 0015 0019
0400	0011 0012 0013 0014 0015
0401	0011 0012 0013 0014 0015
0402	0011 0012 0013 0014 0015 0019
0403	0011 0012 0013 0014 0015 0016
0404	0011 0012 0013 0014 0015 0019
0405	0011 0012 0013 0014 0015
0406	0005 0011 0013 0014 0015
0407	0011 0012 0013 0014 0015 0016
0408	0011 0012 0013 0014 0015 0019
0409	0011 0012 0013 0014 0015
0410	0011 0012 0013 0014 0015 0018
0411	0011 0012 0013 0014 0015 0019
0412	0011 0012 0013 0014 0015 0016
0413	0011 0012 0013 0014 0015 0019
0414	0011 0012 0013 0014 0015 0019
0415	0011 0012 0013 0014 0015
0416	0011 0012 0013 0014 0015
0417	0008 0011 0012 0013 0014 0015
0418	0002 0003 0004 0005 0006
0419	0002 0003 0004 0005

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0420	0002 0003 0004 0005 0006
0421	0002 0003 0004 0005
0422	0002 0003 0004 0005 0009
0423	0002 0003 0004 0005 0008
0424	0002 0003 0004 0005 0006
0425	0002 0003 0004 0005 0006
0426	0002 0003 0004 0005 0006
0427	0002 0003 0004 0005 0006
0428	0002 0003 0004 0005 0006
0429	0002 0003 0004 0005 0008
0430	0002 0003 0004 0005 0006
0431	0002 0003 0004 0005 0006
0432	0002 0003 0004 0005 0006
0433	0002 0003 0004 0005 0009
0434	0002 0003 0004 0005 0009
0435	0002 0003 0004 0005
0436	0002 0003 0004 0005
0437	0002 0003 0004 0005 0006
0438	0002 0003 0004 0005 0006
0439	0002 0003 0004 0005 0006
0440	0002 0003 0004 0005 0006
0441	0002 0003 0004 0005 0006
0442	0002 0003 0004 0005 0006
0443	0002 0003 0004 0005
0444	0002 0003 0004 0005 0006
0445	0002 0003 0004 0005 0009
0446	0002 0003 0004 0005 0009
0447	0002 0003 0004 0005 0009
0448	0002 0003 0004 0005 0006

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0449	0002 0003 0004 0005 0008
0450	0002 0003 0004 0005 0006
0451	0002 0003 0004 0005 0006
0452	0002 0003 0004 0005 0006
0453	0002 0003 0004 0005 0009
0454	0002 0003 0004 0005
0455	0002 0003 0004 0005
0456	0002 0003 0004 0005 0006
0457	0002 0003 0004 0005 0006
0458	0002 0003 0004 0005
0459	0002 0003 0004 0005
0460	0002 0003 0004 0005 0009
0461	0002 0003 0004 0005
0462	0002 0003 0004 0005 0009
0463	0002 0003 0004 0005 0006
0464	0002 0003 0004 0005
0465	0002 0003 0004 0005 0006
0466	0002 0003 0004 0005 0006
0467	0002 0003 0004 0005 0006
0468	0002 0003 0004 0005
0469	0002 0003 0004 0005
0470	0002 0003 0004 0005 0006
0471	0002 0003 0004 0005
0472	0002 0003 0004 0005 0008
0473	0002 0003 0004 0005
0474	0002 0003 0004 0005
0475	0002 0003 0004 0005
0476	0002 0003 0004 0005 0006
0477	0002 0003 0004 0005 0006



<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0478	0002 0003 0004 0005
0479	0002 0003 0004 0005 0009
0480	0002 0003 0004 0005
0481	0002 0003 0004 0005
0482	0002 0003 0004 0005
0483	0002 0003 0004 0005
0484	0002 0003 0004 0005 0006
0485	0002 0003 0004 0005 0009
0486	0001 0002 0003 0004 0005
0487	0001 0002 0003 0004 0005
0488	0002 0003 0004 0005 0006
0489	0002 0003 0004 0005 0009
0490	0002 0003 0004 0005
0491	0001 0002 0003 0004 0005
0492	0002 0003 0004 0005
0493	0002 0003 0004 0005
0494	0002 0003 0004 0005 0009
0495	0002 0003 0004 0005 0006
0496	0002 0003 0004 0005 0006
0497	0002 0003 0004 0005
0498	0002 0003 0004 0005
0499	0002 0003 0004 0005
0500	0002 0003 0004 0005
0501	0002 0003 0004 0005 0006
0502	0002 0003 0004 0005 0006
0503	0002 0003 0004 0005 0009
0504	0002 0003 0004 0005 0006
0505	0002 0003 0004 0005 0006
0506	0002 0003 0004 0005 0006

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0507	0002 0003 0004 0005 0008
0508	0002 0003 0004 0005 0009
0509	0002 0003 0004 0005 0009
0510	0002 0003 0004 0005 0009
0511	0002 0003 0004 0005
0512	0002 0003 0004 0005
0513	0002 0003 0004 0005 0006
0514	0002 0003 0004 0005 0006
0515	0002 0003 0004 0005
0516	0002 0003 0004 0005 0006
0517	0002 0003 0004 0005 0006
0518	0002 0003 0004 0005 0009
0519	0002 0003 0004 0005 0006
0520	0002 0003 0004 0005 0006
0521	0002 0003 0004 0005
0522	0002 0003 0004 0005
0523	0002 0003 0004 0005 0009
0524	0002 0003 0004 0005 0009
0525	0002 0003 0004 0005
0526	0002 0003 0004 0005 0008
0527	0002 0003 0004 0005 0006
0528	0002 0003 0004 0005 0006
0529	0002 0003 0004 0005
0530	0002 0003 0004 0005
0531	0002 0003 0004 0005 0006
0532	0002 0003 0004 0005
0533	0002 0003 0004 0005
0534	0002 0003 0004 0005
0535	0002 0003 0004 0005 0006

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0536	0002 0003 0004 0005 0006
0537	0002 0003 0004 0005
0538	0002 0003 0004 0005 0006
0539	0002 0003 0004 0005
0540	0002 0003 0004 0005
0541	0002 0003 0004 0005
0542	0002 0003 0004 0005
0543	0002 0003 0004 0005
0544	0002 0003 0004 0005
0545	0002 0003 0004 0005 0009
0546	0002 0003 0004 0005 0006
0547	0002 0003 0004 0005
0548	0002 0003 0004 0005 0008
0549	0002 0003 0004 0005
0550	0002 0003 0004 0005
0551	0002 0003 0004 0005 0009
0552	0002 0003 0004 0005
0553	0002 0003 0004 0005 0009
0554	0002 0003 0004 0005
0555	0002 0003 0004 0005 0006
0556	0002 0003 0004 0005
0557	0002 0003 0004 0005
0558	0001 0002 0003 0004 0005
0559	0002 0003 0004 0005 0009
0560	0002 0003 0004 0005
0561	0002 0003 0004 0005
0562	0002 0003 0004 0005 0006
0563	0002 0003 0004 0005 0006
0564	0002 0003 0004 0005

11-22

<b>Mahasiswa</b>	<b>Mata Kuliah</b>
0565	0002 0003 0004 0005 0006
0566	0002 0003 0004 0005 0006
0567	0002 0003 0004 0005

## LAMPIRAN 2

Lampiran ini merupakan hasil pemetaan dari data matakuliah setelah dilakukan praproses. Data ini disimpan pada file .crs

<b>Mata Kuliah</b>	<b>Jumlah Mahasiswa</b>
0001	4
0002	155
0003	150
0004	154
0005	152
0006	64
0007	157
0008	17
0009	28
0010	1
0011	160
0012	150
0013	161
0014	159
0015	144
0016	21
0017	190
0018	176
0019	138
0020	172
0021	184
0022	191
0023	28
0024	30
0025	27

12-2

<b>Mata Kuliah</b>	<b>Jumlah Mahasiswa</b>
0026	28
0027	40
0028	4
0029	18
0030	50
0031	38
0032	40

### LAMPIRAN 3

Lampiran ini merupakan hasil usulan jadwal UTS hasil luaran algoritma *greedy simulated annealing*

**Tabel 13.1 Usulan Jadwal UTS Hasil Simulated Annealing**

Hari	Sesi	Kode	Matakuliah
Senin	1	KS141210	Tata Tulis Ilmiah
		KS141307	Interaksi Manusia dan Komputer
	2	KS141205	Algoritma dan Struktur Data
		KS141407	Pemrograman Integratif
		KS141411	Forensika Digital
	3	KS141318	Manajemen Resiko
Selasa	1	KS141304	Pengantar Basis Data
		KS141310	Pemrograman Berbasis Web
		KS141403	Riset Operasi Lanjut
	2	KS141207	Arsitektur SI/TI Perusahaan
		KS141316	Manajemen Layanan TI

Hari	Sesi	Kode	Matakuliah
		KS141402	Penggalian Data dan Analitika Bisnis
		KS141413	Manajemen Kualitas SI/TI
	3	KS141317	Tata Kelola TI
Rabu	1	KS141308	Keamanan Aset Informasi
	2	KS141204	Bahasa Pemrograman
		KS141209	Statistika
		KS141404	Teknik Peramalan
	3	KS141303	Dasar-Dasar Pengembangan Perangkat Lunak
		KS141321	Manajemen dan Administrasi Basis Data
Kamis	1	KS141309	Desain Basis Data
		KS141323	Pengukuran dan Evaluasi TI
		KS141405	Pemrograman Perangkat Bergerak
	2	KS141208	Kepemimpinan dan Ketrampilan Interpersonal
		KS141325	Etika Profesi SI
	3	KS141301	Desain dan Manajemen Proses Bisnis
		KS141322	Sistem Cerdas



Hari	Sesi	Kode	Matakuliah
Jumat	1	KS141306	Analisis dan Desain Perangkat Lunak
		KS141409	E-Business
	2	KS141206	Pengantar Sistem Operasi
		KS141319	Manajemen Pengadaan dan Investasi TI
	3	KS141320	Man. Rantai Pasok dan Hub. Pelanggan

*Halaman ini sengaja dikosongkan*

## LAMPIRAN 4

Lampiran ini merupakan hasil usulan jadwal UAS dengan 15 slot ujian hasil luaran algoritma *greedy simulated annealing*

**Tabel 14.1 Usulan Jadwal UAS Hasil Simulated Annealing**

Hari	Sesi	Kode	Matakuliah
Senin	1	KS141205	Algoritma dan Struktur Data
		KS141319	Manajemen Pengadaan dan Investasi TI
	2	KS141322	Sistem Cerdas
	3	KS141310	Pemrograman Berbasis Web
		KS141323	Pengukuran Kinerja dan Evaluasi TI
		KS141404	Teknik Peramalan
Selasa	1	KS141207	Arsitektur SI/TI Perusahaan
		KS141316	Manajemen Layanan TI
		KS141325	Etika Profesi
	2	KS141317	Tata Kelola TI

Hari	Sesi	Kode	Matakuliah
	3	KS141409	E-Business
		KS141306	Analisis dan Desain Perangkat Lunak
Rabu	1	KS141206	Pengantar Sistem Operasi
		KS141407	Pemrograman Integratif
		KS141413	Manajemen Kualitas SI/TI
	2	KS141301	Desain dan Manajemen Proses Bisnis
		KS141318	Manajemen Resiko TI
	3	KS141309	Desain Basis Data
		KS141402	Penggalian Data dan Analitika Bisnis
		KS141403	Riset Operasi Lanjut
Kamis	1	KS141204	Bahasa Pemrograman
		KS141209	Statistika
		KS141411	Forensika Digital
	2	KS141303	Dasar-Dasar Pengembangan Perangkat Lunak
		KS141320	Manajemen Rantai Pasok dan Hub. Pelanggan
		KS141405	Pemrograman Perangkat Bergerak

Hari	Sesi	Kode	Matakuliah
	3	KS141308	Keamanan Aset Informasi
Jumat	1	KS141208	Kepemimpinan dan Keterampilan Interpersonal
		KS141321	Manajemen dan Administrasi Basis Data
	2	KS141210	Tata Tulis Ilmiah
		KS141304	Pengantar Basis Data
		KS141307	Interaksi Manusia dan Komputer
	3		